Research in Astronomy and Astrophysics, 22:045014 (7pp), 2022 April https://doi.org/10.1088/1674-4527/ac56cd

Performance Evaluation of Baseline-dependent Averaging Based on Full-scale SKA1-LOW Simulation

Qing-Wen Deng[1,2], Feng Wang[1,2], Hui Deng[1,2], Ying Mei[1,2], Jing Li[1,2], Oleg Smirnov[3], and Shao-Guang Guo[4]
[1] Center for Astrophysics, School of Physics and Materials Science, Guangzhou University, Guangzhou 51006, China; fengwang@gzhu.edu.cn
[2] Great Bay Center, National Astronomical Data Center, Guangzhou, 510006, China
[3] Department of Physics and Electronics, Rhodes University, PO Box 94, Makhanda, 6140, South Africa
[4] Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai, 200030, China

Received 2021 November 22; revised 2022 February 14; accepted 2022 February 16; published 2022 March 17

Abstract

The Square Kilometre Array (SKA) is the largest radio interferometer under construction in the world. Its immense amount of visibility data poses a considerable challenge to the subsequent processing by the science data processor (SDP). Baseline dependent averaging (BDA), which reduces the amount of visibility data based on the baseline distribution of the radio interferometer, has become a focus of SKA SDP development. This paper developed and implemented a full-featured BDA module based on Radio Astronomy Simulation, Calibration and Imaging Library (RASCIL). Simulated observations were then performed with RASCIL based on a full-scale SKA1-LOW configuration. The performance of the BDA was systematically investigated and evaluated based on the simulated data. The experimental results confirmed that the amount of visibility data is reduced by about 50% to 85% for different time intervals ($\Delta t_{\max}$). In addition, different $\Delta t_{\max}$ have a significant effect on the imaging quality. The smaller $\Delta t_{\max}$ is, the smaller the degradation of imaging quality.

Key words: instrumentation: radio interferometers – methods: analytical mathematics – techniques: astronomical simulations

1. Introduction

The Square Kilometre Array (SKA) (Braun 1996) is an ongoing international project to build the world's largest radio interferometric telescope with more than one square kilometer potential collection area. With detailed design and preparation well underway, the SKA represents a giant leap forward in engineering to deliver a unique instrument.

The *uv* distribution of a radio interferometer generally has a dense center and a sparse edge. With the rotation of the Earth, each sampling point of the radio interferometer draws an arc-shaped trajectory around the phase center on the *uv* plane. Short baselines have denser data than long baselines for the same track length, and the difference can be even greater for a larger array. Decorrelation can be avoided on the longer baselines when more samples are averaged at the center than at the outer edges. At the same time, data compression can be carried out on shorter baselines.

Baseline dependent averaging (BDA) was proposed by Cotton (1986, 1999) to reduce the visibility data volumes, which has been used by the Murchison Widefield Array (MWA, Mitchell et al. 2008) and is also used to shape the field of interest (Atemkeng et al. 2018). However, averaging visibilities over time and frequency will cause image distortion, also called the smearing effect. Bandwidth smearing is manifested as a position-dependent and radial convolution effect in the image field. Time smearing is similar to bandwidth smearing but more complicated, described as a loss in amplitude (Cotton 1986, 1999). Bridle & Schwab (1999) conducted a mathematical analysis of these two smearing effects and found that they cannot be effectively corrected by calibration or self-calibration methods. It is recommended to design a comprehensive observation strategy to reduce the impact to an acceptable level. Therefore, the short integration time and small channel width are necessary for the long baselines utilized in a radio interferometer to suppress the smearing effect. In contrast, the resolution requirement for time and frequency is relatively low on short baselines.

The BDA does not change the channel bandwidth and integration time for long baseline visibility data, but averaging of short baseline visibility data corresponds to an increase in integration time and channel bandwidth. Wijnholds et al. (2018) obtained the Cramer–Rao bound of averaged visibilities by estimating the number of raw visibilities and comparing it with the covariance obtained by the error transfer formula. It is proven that BDA will not cause other effects except for the approximately obtainable decorrelation loss. Salvini & Wijnholds (2017) proposed the Compress-Expand-Compress method to expand the visibilities to the required time resolution for calibration after the first compression, and then perform the second compression, and finally achieve a high compression ratio (CR) of the visibilities in time. However, few previous literature works presented the quantitative analysis of BDA on the final image quality and storage costs.

As the SKA enters its construction phase and the SKA Regional Centers (SRCs) construction white paper has been released, it becomes imperative to research the BDA technique further and analyze its usability for the SKA1 scale. We wish to analyze and discuss this study systematically: 1. How much space would be saved by using BDA technology for SKA1-LOW observations? 2. Is there a significant degradation of the final dirty image with the BDA?

In the rest of this study, we first introduce the BDA algorithm and its implementation. We then simulate full-scale SKA1-LOW observations and investigate the BDA performance in Section 3. The discussions are described in Section 4. The conclusions and future work are presented in the last section.

2. The Algorithm and Implementation of BDA

2.1. The BDA Algorithm

For a radio interferometer, a visibility function is obtained by correlating the signal collected by two antennas of each baseline with the same time interval δt and frequency sampling interval δf. According to the mathematical definition of BDA (Wijnholds et al. 2018), we can average the raw visibilities and thus obtain the averaged visibilities. In data processing, from P receiving antennas P^2 correlations are assumed to be collected in K short-term integrations, either over time, frequency, or both. The raw visibility data vector can be defined as

$$\boldsymbol{r} = [r_1,\cdots,r_K]^T \in \mathbb{C}^{KP^2\times 1}, \tag{1}$$

where $\mathbb{C}$ denotes the complex matrix. The averaging process can be described as

$$\boldsymbol{r}_{\text{ave}} = \boldsymbol{W}\boldsymbol{I}_{\text{s}}\boldsymbol{r} \in \mathbb{C}^{M\times 1}, \tag{2}$$

where $\boldsymbol{W} = (\boldsymbol{I}_{\text{s}}\boldsymbol{I}_{\text{s}}^T)^{-1}$ is a diagonal weighting matrix. $\boldsymbol{I}_{\text{s}} \in \mathbb{R}^{M\times KP^2}$ is a sparse matrix that determines which samples will be averaged together, where $\mathbb{R}$ denotes the real matrix and M is the length of the average visibility data vector.

Suppose we ignore the correlation effects and assume that the values of the visibility data averaged together are the same. In that case, the raw visibilities can be obtained approximately from the averaged visibilities by

$$\boldsymbol{r} = \boldsymbol{I}_{\text{s}}^T\boldsymbol{r}_{\text{ave}}. \tag{3}$$

The selection matrix $\boldsymbol{I}_{\text{s}}$ is related to the averaging intervals of time and frequency in the BDA. For a selected baseline D, the averaging intervals can be calculated by the rounding ratio of that baseline to the longest baseline $B_{\max}$ as

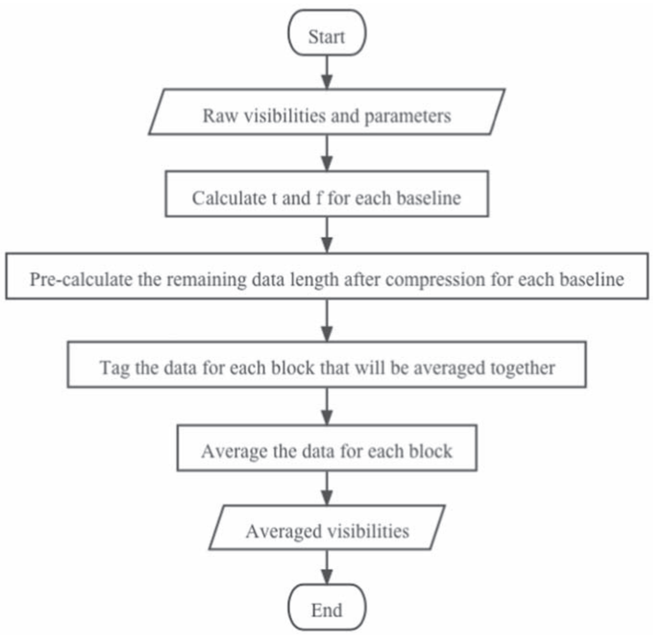
Start
Raw visibilities and parameters
Calculate t and f for each baseline
Pre-calculate the remaining data length after compression for each baseline
Tag the data for each block that will be averaged together
Average the data for each block
Averaged visibilities
End

Figure 1. The flow chart of the BDA implementation.

$$t = \delta t\frac{B_{\max}}{D}, \quad f = \delta f\frac{B_{\max}}{D}. \tag{4}$$

We used the rounding ratio of $B_{\max}$ to D to determine whether BDA processing is needed. On partial long baselines, when $\frac{B_{\max}}{D} = 1$, the data will not be averaged. While on the short baselines, $\frac{B_{\max}}{D} > 1$, indicating that more sampling data can be averaged. In addition, t is limited by the calibration timescale determined by the environment and instrument for the interferometer. A larger averaging scale will also make the smearing effects more serious. Therefore, it is necessary to set reasonable upper limits for t and f in the implementation.

2.2. Implementation

We implemented a full-featured BDA module based on the Radio Astronomy Simulation, Calibration and Imaging Library (RASCIL).[5] RASCIL is a pure Python software package suite for radio interferometer calibration and imaging algorithms, especially for SKA data processing. Since the public release of RASCIL, it has been widely used in data processing for some radio interferometers (e.g., Wei et al. 2021 and so on). We developed a BDA module based on RASCIL, which has been released online.[6]

The flowchart of the implementation is displayed in Figure 1. Figure 2 shows an example of the averaging process on a

[5] https://gitlab.com/ska-telescope/external/rascil

[6] https://github.com/astronomical-data-processing/ska-bda

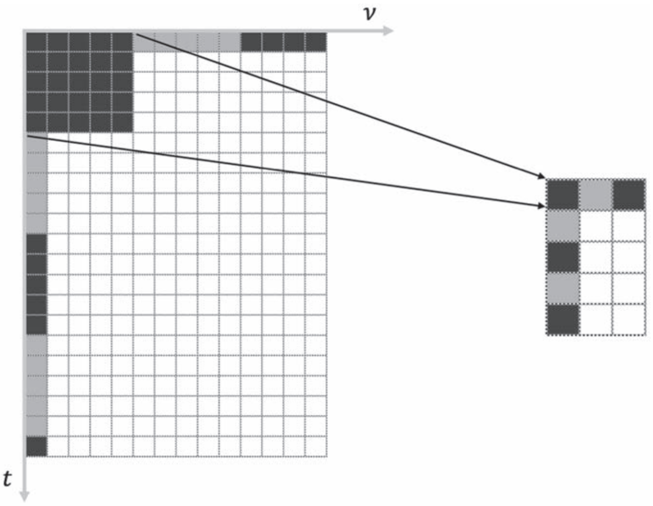

Figure 2. The averaging of visibility data at a baseline when applying BDA.

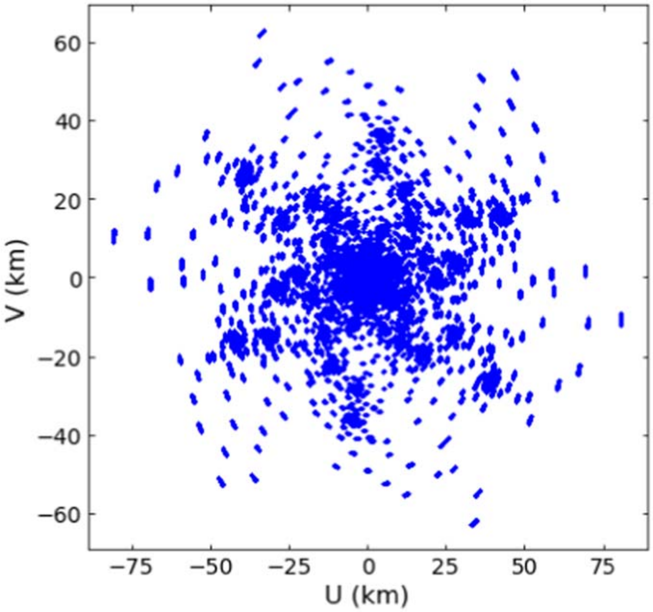

Figure 3. The *uv* distribution of the SKA1-LOW observed in a single channel at 100 MHz for 12 minutes.

baseline, where the data will be reduced from the original 21 × 14 to 5 × 3, by assuming that both t and f are 5 according to Equation (4). In the case of the shortest baselines, t and f are also equal to the upper limits of the time-frequency interval used in the BDA, defined as $\Delta t_{\max}$ and $\Delta f_{\max}$. In the averaging process, we first calculate the position of the raw visibilities corresponding to the flattened averaged visibilities. We then average the visibilities based on the positions and number of visibilities.

We used the BlockVisibility class defined in RASCIL. The visibility data were stored using a multi-dimensional array, with dimensions including baseline, polarization, time and frequency. To meet the requirements of BDA performance profiling, we implemented BDA by using three underlying packages, i.e., pure Python, Pandas and Numba, respectively.

In the pure Python implementation, we grouped the data for averaging based on Numpy. To optimize the performance, we tried to use Pandas, put all the parameter data into a table when preprocessing and then performed group-by operations to complete all the calculations.

We also used Numba (Lam et al. 2015) to speed up the function and further improved the processing performance. Numba is an on-the-fly compiler that translates a subset of the Python and Numpy code used in the function into efficient machine code, which can effectively improve the speed of the program.

3. Performance Assessments For BDA

3.1. Observational Configuration

To more accurately evaluate the performance of the BDA, we used RASCIL to simulate single-channel and one polarization visibility data observed by the full scale SKA1-LOW telescope. During simulation, we use all SKA1-LOW 512 stations and set up 12 minutes of observations, of which 6 minutes were on each side of the zenith. The integration time is set to 0.9 s as required by the array structure. The observing frequency is 100 MHz with the channel bandwidth of 1 MHz, and the phase center in the observation points to R.A. 15° and decl. −45°. We finally obtained 800 temporal sampling points on each baseline. The *uv* distribution of the simulated observation is shown in Figure 3.

3.2. Observation Simulation

With the observational configuration described above, we simulated observations of point and extended sources separately. We selected the corresponding sources from the GaLactic and Extragalactic All-sky Murchison Widefield Array (GLEAM) survey catalog (Hurley-Walker et al. 2017) with an imaging size of 32,768 × 32,768 pixels. The other is the M31 image that is observed by the Very Large Array. The image has a pixel size of 512 × 512 and a resolution of 1 arcsecond. We used the transform.resize() function in skimage (van der Walt et al. 2014) to scale the M31 image to the same pixel scale as the image generated by the GLEAM model for this study. It should be clear here that such a magnification of the original image is only necessary for the simulation of the observation. Two dirty images for the cases are shown in Figure 4.

3.3. Evaluation Results

We invoke the BDA module to perform visibility compression, decompression and imaging processing. Since BDA

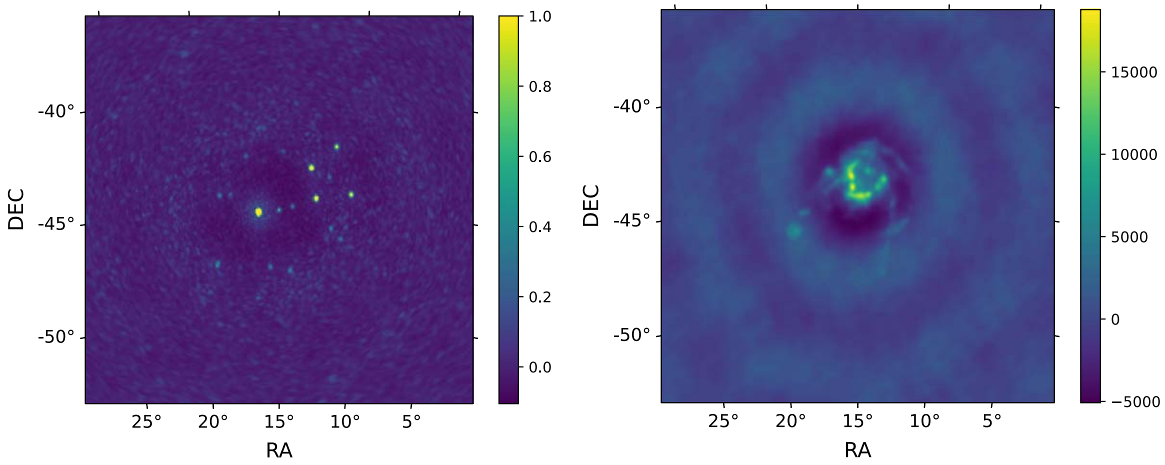

Figure 4. Dirty images of the raw visibilities observed from the GLEAM model and the M31 model images.

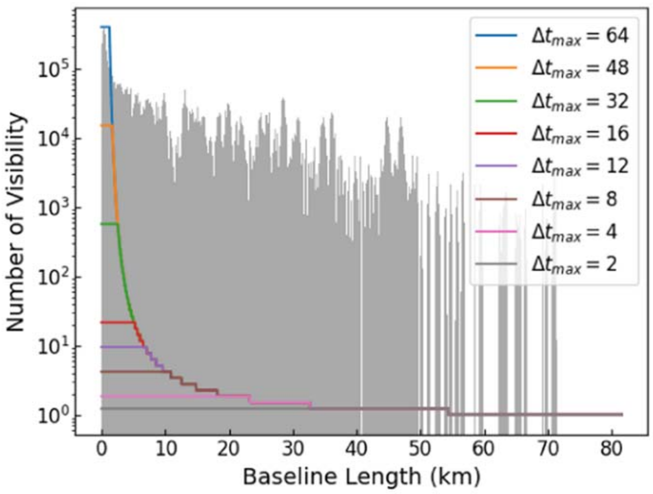

Figure 5. The distribution of visibilities with baseline length, and the effect curves of the CR.

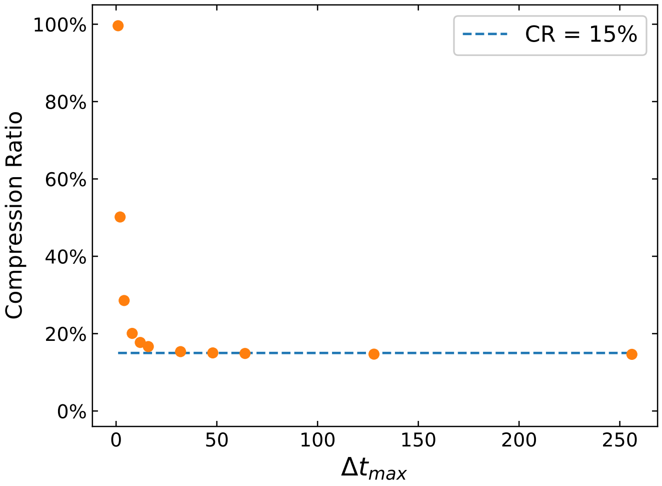

Figure 6. The trend chart of the CR with the maximum number of samples in averaging $\Delta t_{\max}$.

processing at frequency series yields similar results as time series, we only simulate and analyze temporal BDA in this study. Also, the effects on visibility data and dirty images are investigated by setting different upper limits on time integration in the BDA.

3.3.1. The Compression Ratio

The CR is calculated by comparing the data volumes between the averaged and raw visibilities, as CR $= \frac{r_{\text{ave}}}{r} \times 100\%$.

We set different upper limits ($\Delta t_{\max}$), i.e., 1, 2, 4, 8, 12, 16, 32, 48, 64, 128 and 256, to evaluate the CR of the BDA. $\Delta t_{\max}$ also means the maximum number of samples being averaged together on the shortest baselines. These different upper limits lead to different CRs over the baseline length range, and some of these variation curves are shown in Figure 5. Since the shorter baselines have larger data volumes, we want to average the amount of data over the shorter baselines as much as possible. We can also find that the increase of $\Delta t_{\max}$ only further compresses the data on the shorter baselines, while the volume proportion of these data is decreasing in the total.

The final result of the CR with different $\Delta t_{\max}$ is displayed in Figure 6. With the increasing value of $\Delta t_{\max}$, the CR reduces quickly and then becomes slow. Finally, a larger $\Delta t_{\max}$ does not significantly improve the final CR. It changes very little after the CR reaches 15%, where $\Delta t_{\max} = 48$. To avoid the more severe errors that may arise from a bigger $\Delta t_{\max}$, it is worth considering using a $\Delta t_{\max}$ less than 48 in subsequent studies.

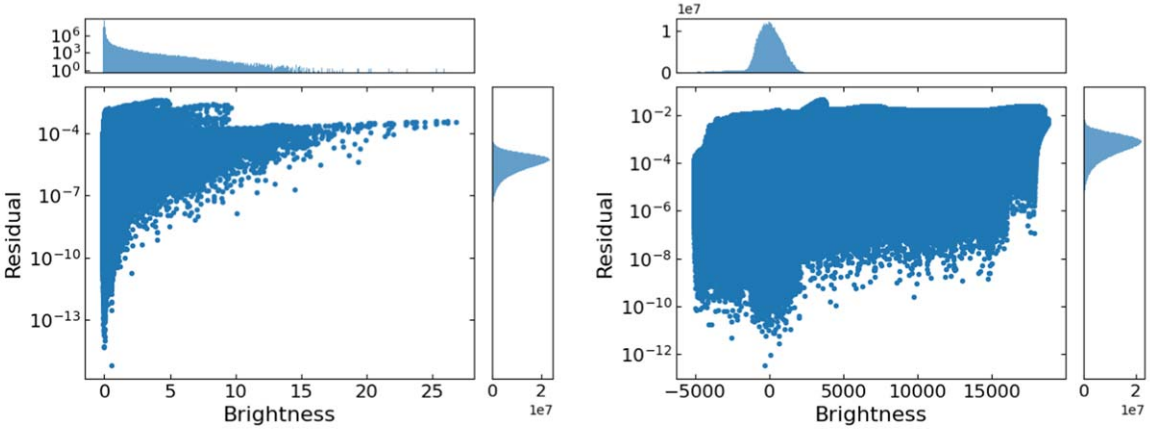

Figure 7. The distribution of the absolute value of the residual with brightness, corresponding to the dirty image results of GLEAM (left) and M31 (right) as observed model images, and $\Delta t_{\max} = 48$.

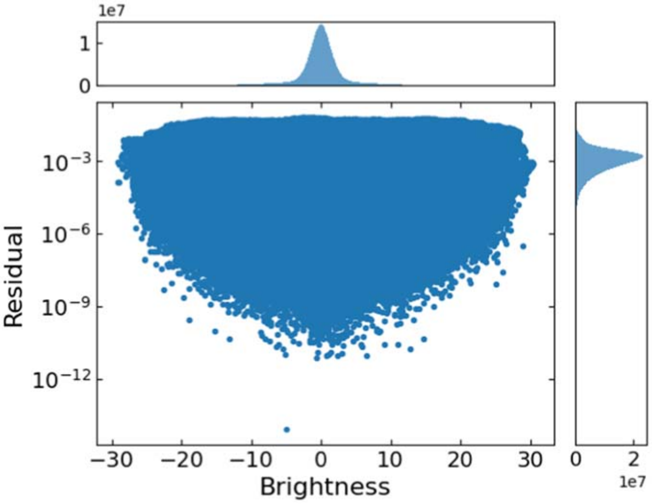

Figure 8. Distribution of the absolute value of the residual with brightness for the pure noise model results, and $\Delta t_{\max} = 48$.

3.3.2. Imaging Quality Evaluation

In addition to the CR, the impact on the image quality after applying BDA is a significant issue. We defined the visibility data of the simulated observations as raw visibilities, the visibility data processed by the BDA as the averaged visibilities and the final decompressed visibility data as recovered visibilities. We first decompressed the averaged visibilities by using a method similar to linear interpolation and then used the recovered visibilities for the subsequent image processing.

To exclude the possible effects of different deconvolution methods on the imaging results, we used dirty images to analyze the imaging quality. Due to the difference between the recovered visibilities and the raw visibilities, the brightnesses in the dirty images are not the same. The deviations may be positive or negative relative to the dirty image of the raw visibilities. For convenience, the absolute value of the deviation is used here, and its distribution with the brightness is displayed in Figure 7, where $\Delta t_{\max}$ of the recovered visibilities is equal to 48.

The brightness distribution of the dirty images is mainly concentrated around 0, and the lower limit of deviation increases with brightness, but the upper limit does not change excessively. At the same time, the maximum value of the residuals is small.

We used a pure noise image as a sky model for simulated observations and performed the same BDA processing. In generating this noisy model image, the same image size as the previous model was used, filled only with Gaussian noise with a mean of 0 and a standard deviation of 0.1. Figure 8 shows the result of the dirty image when $\Delta t_{\max}$ is equal to 48, where the maximum error is 0.0698. Moreover, when $\Delta t_{\max}$ is equal to 2, the maximum error is 0.0162. During this experiment, we tried to reduce the overall amplitude of the noisy model image by a certain ratio. The corresponding change in the dirty image was that both deviation and brightness values were reduced by the same ratio, while the contours of Figure 8 did not change much.

The statistical distribution of the residuals exhibits a Gaussian-like distribution in the dirty image results when displayed in logarithmic form. The mean and standard deviation of the residuals in this form are expressed in Table 1, where Case 1 refers to the results of dirty images for GLEAM, and Case 2 refers to the ones for M31.

The standard deviations of these two cases are approximately the same and do not change significantly with $\Delta t_{\max}$. This indicates that the change in residual is more like an overall shift, while the mean is the distance of the shift.

Figure 9 plots the relationship between the means and $\Delta t_{\max}$ for these two cases, fitted with a logarithmic function for each. The same is that a small $\Delta t_{\max}$ corresponds to a small imaging

Table 1
Mean and Standard Deviation of the Logarithmic Residual in Two Cases

		Case 1			Case 2		
$\Delta t_{\max}$	CR	Mean	Std.	Max	Mean	Std.	Max
2	50.16%	−6.5449	0.5857	−3.1033	−7.6400	0.5309	−5.4398
4	28.56%	−6.0538	0.5570	−2.6526	−6.6431	0.5292	−4.4348
8	20.06%	−5.8964	0.5514	−2.5298	−5.7456	0.5250	−3.6300
12	17.72%	−5.8079	0.5485	−2.4826	−5.1231	0.5366	−3.0849
16	16.66%	−5.7466	0.5460	−2.4582	−4.8532	0.5250	−2.7622
32	15.35%	−5.6147	0.5401	−2.4191	−3.9763	0.5264	−1.9117
48	15.02%	−5.5350	0.5378	−2.3999	−3.3577	0.5440	−1.3557

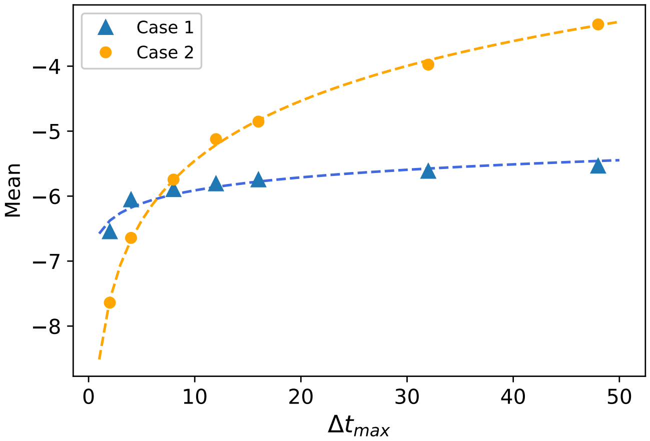

Figure 9. The trend of the different mean values with $\Delta t_{\max}$ in two groups.

error, while case 2 has a larger variation range of the means than case 1. This difference is probably due to the different characteristics of the amplitude intensity distribution of visibilities on the *uv* plane in these two cases. Case 1 is relatively uniform, while case 2 is more concentrated in the low-frequency part.

Considering both the CR and dirty image quality, a small $\Delta t_{\max}$ (e.g., $\Delta t_{\max} = 12$) could meet the requirements for common use. We also found that further compression over short baselines is the cause of the relative error in dirty images. A large $\Delta t_{\max}$ implies large deviations in the recovered visibilities on short baselines. In practice, it is difficult to invert a suitable $\Delta t_{\max}$ from the imaging results, while choosing a small one is feasible and safe.

3.4. The Processing Performance

The processing performance of BDA is a fundamental metric. A series of tests was performed on a Centos 7 server equipped with 32 processors (Intel Xeon Gold 6226R), 2.9 GHz core frequency and 1024 GB of RAM. The version of RASCIL utilized to obtain the simulation data in the tests was v.0.1.11. Using 12 minute single-channel simulation data of SKA1-LOW with a data volume of 12.5 GB, we profile the BDA module optimized by Numba, pure Python code and Pandas. The performance results are presented in Table 2. The BDA module implemented using Numba has the best performance.

Table 2
The Performance of the BDA Implementation with Different Data Volumes

Data volume (GB)	Numba (s)	Pandas (s)	Pure Python (s)
1.56	10.75	13.69	400.01
3.13	17.48	23.78	697.62
6.25	29.02	44.21	1413.21
12.50	58.79	86.18	2941.89
25.00	105.59	171.38	6523.35
50.00	243.12	367.48	16 178.32

Therefore, we further tested the processing performance of the Numba optimized code with a series of simulated data of different observation times and four channels. The maximum observation time was 48 minutes and the data volume obtained was 143.75 GB. During the process of BDA, $\Delta t_{\max}$ was set to 6, 12 and 24, and $\Delta f_{\max}$ was 1. The performance result is depicted in Figure 10, fitted with an exponential function.

From the results shown in Figure 10, the time consumption of BDA processing is essentially linear with the amount of data to be processed. The processing speed of BDA in the case of a single process is about 13 GB per minute. Further improvements are expected under parallel computing conditions. This speed is acceptable for data pre-processing at SRCs. In addition, the processing speed has little to do with the amount of $\Delta t_{\max}$.

4. Discussions

Experimental results indicated a significant decrease in the capacity of the visibility data with different $\Delta t_{\max}$, which indicates that BDA is very valuable for SKA data processing,

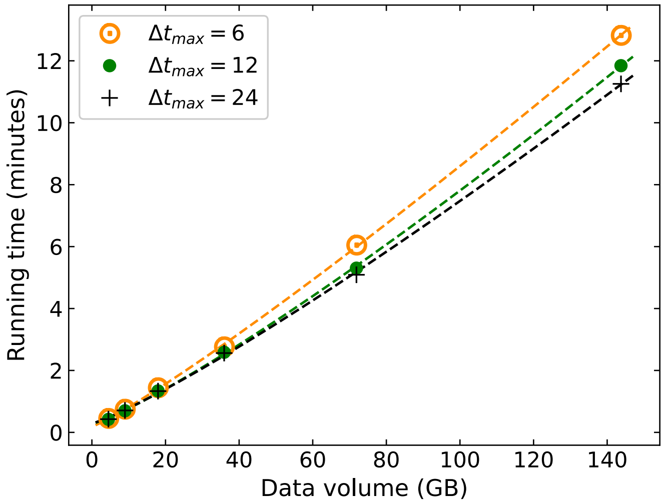

Figure 10. The performance of the BDA implementation optimized with Numba at different data volumes.

especially for the construction of subsequent SRCs. The annual storage capacity of SRCs will be at least 5 petabytes (PB) per year in the beginning, which will increase to at least 1.7 exabytes by 2028, with at least 700 PB online (Bolton 2019). BDA can compress at least 50% of the short baseline visibility data, which is valuable for reducing the cost of SRC construction.

This study also examined the variation of the imaging quality at different $\Delta t_{\max}$. The experimental results provided an essential reference for SKA1-LOW to carry out Epoch of Reionization and cosmic dawn research. The MWA data were averaged (Mitchell et al. 2008), but no specific details were given.

5. Conclusions

We implemented a new BDA module based on RASCIL, and this implementation was created by designing functions using Numba. It has not introduced excessive memory usage in the tests and completes the computation tasks faster than the other modules. The speed of a single process is around 13 GB per minute. It also performed well during the BDA processing of the simulation data.

Through the simulation of observing the GLEAM and M31 model images with SKA1-LOW, we evaluate the performance of BDA. According to the subsequent analysis, the error due to BDA increases with the maximum upper limit of the averaging interval on short baselines. In contrast, the CR does not improve all the time, and the reduction in data volume remains at a maximum ratio of approximately 85%. The smaller upper limit is sufficient for the CR, and the imaging error is reasonable. Overall, the BDA technology will have applications in the face of massive SKA observation data processing. The BDA can effectively reduce the storage space of visibility data, as it is also valuable for the future construction and application of SRCs.

Acknowledgments

This work is supported by the National SKA Program of China (2020SKA0110300), the Joint Research Fund in Astronomy (U1831204, U1931141) under cooperative agreement between the National Natural Science Foundation of China (NSFC) and the Chinese Academy of Sciences (CAS), the Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (11961141001) and the National Science Foundation for Young Scholars (11903009). This work is also supported by the Astronomical Big Data Joint Research Center, co-founded by National Astronomical Observatories, Chinese Academy of Sciences and Alibaba Cloud.

We sincerely appreciate the anonymous referee for valuable and helpful comments and suggestions.

References

Atemkeng, M., Smirnov, O., Tasse, C., et al. 2018, MNRAS, 477, 4511

Bolton, R. C. 2019, SKA Regional Centre Requirements, Technical Report, SKA-TEL-SKO-0000735, internal SKA document

Braun, R. 1996, in Astrophysics and Space Science Library, The Westerbork Observatory, Continuing Adventure, ed. E. Raimond & R. Genee, Vol. 208 (Dordrecht: Springer), 167

Bridle, A. H., & Schwab, F. R. 1999, in ASP Conf. Ser. 180, Synthesis Imaging in Radio Astronomy II, ed. G. B. Taylor, C. L. Carilli, & R. A. Perley (San Francisco, CA: ASP), 371

Cotton, W. D. 1986, in Synthesis Imaging, ed. R. A. Perley, F. R. Schwab, & A. H. Bridle (Green Band: NRAO), 123

Cotton, W. D. 1999, in ASP Conf. Ser. 180, Synthesis Imaging in Radio Astronomy II, ed. G. B. Taylor, C. L. Carilli, & R. A. Perley (San Francisco, CA: ASP), 357

Hurley-Walker, N., Callingham, J. R., Hancock, P. J., et al. 2017, MNRAS, 464, 1146

Lam, S. K., Pitrou, A., & Seibert, S. 2015, in 2nd Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15 (New York: Association for Computing Machinery), 15

Mitchell, D. A., Greenhill, L. J., Wayth, R. B., et al. 2008, ISTSP, 2, 707

Salvini, S., & Wijnholds, S. J. 2017, in 2017 XXXIInd General Assembly and Scientific Symp. of the International Union of Radio Science (URSI GASS) (Montreal: IEEE), 1

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., et al. 2014, arXiv:1407.6245

Wei, S.-L., Luo, K.-D., Wang, F., Deng, H., & Mei, Y. 2021, RAA, 21, 080

Wijnholds, S. J., Willis, A. G., & Salvini, S. 2018, MNRAS, 476, 2029