Xova: Baseline-Dependent Time and Channel Averaging for Radio Interferometry

Marcellin Atemkeng¹, Simon Perkins², Jonathan Kenyon¹, Benjamin Hugo²,¹, and Oleg Smirnov¹,²

¹Rhodes University, Makhanda (Grahamstown), Eastern Cape, South Africa
²South African Radio Astronomy Observatory, Cape Town, Western Cape, South Africa

Abstract. Xova is a software package that implements baseline-dependent time and channel averaging on Measurement Set data. The $uv$-samples along a baseline track are aggregated into a bin until a specified decorrelation tolerance is exceeded. The degree of decorrelation in the bin correspondingly determines the amount of channel and timeslot averaging that is suitable for samples in the bin. This necessarily implies that the number of channels and timeslots varies per bin and the output data loses the rectilinear input shape of the input data.

1. Effects of Time and Channel Averaging

Consider $V_{pq} = V(u_{pq}(t, ν))$ as the visibility sampled by the baseline $pq$ at time $t$ and frequency $ν$. An interferometer is non-ideal in the sense that the measured visibility is the average of this sampled visibility over the sampling bin, $B_{kr}^{[Δt,Δν]} = [t_k - Δt/2, t_k + Δt/2] × [ν_r - Δν/2, ν_r + Δν/2]$:

$$\overline{V}_{pq} = \frac{1}{ΔtΔν} \int_{B_{kr}^{[Δt,Δν]}} V(u_{pq}(t, ν))dt dν,$$

(1)

where Δ$t$ and Δ$ν$ are the integration intervals. If $Π_{pq}$ represents a normalized 2D top-hat window for a baseline $pq$ then Eq. (1) is equivalent to the infinitesimal integral:

$$\overline{V}_{pq} = \int_\infty \Pi_{pq}(t - t_k, ν - ν_r) V_{pq}(t, ν) dt dν,$$

(2)

which is a convolution in the Fourier space, i.e.:

$$\overline{V}_{pq} = [Π_{pq} \circ V_{pq}](u_{pq}(t_k, ν_r))$$

(3)

$$= δ_{pqkr} [Π_{pq} \circ V_{pq}],$$

(4)

where $δ_{pqkr}(u) = δ(u - u_{pqkr})$ is the Delta function shifted to the sampled point $pqkr$. For an observation with frequency range $F = Δν×N_ν$ and total observing period $T = Δt×N_t$, observing for long times and large frequency ranges leads to storage issues, as well as
computation cost since \( \mathcal{T}, \mathcal{F} \propto (N_t, N_v) \) if \( \Delta t \) and \( \Delta \nu \) most remain sufficiently small. For aggressive averaging, \( \Delta t \) and \( \Delta \nu \) are large which makes \( \Pi_{pq} \) to deviate significantly from \( \delta_{pqkr} \) and therefore causes the visibility to decorrelate: \( \mathcal{V}_{pq} \neq \hat{\mathcal{V}}_{pq} \). To derive the effect of averaging on the image, we can reformulate Eq. 4 as:

\[
\hat{\mathcal{V}}_{pq} = \mathcal{F}(\mathcal{P}_{pqkr}((\Pi_{pq} \circ \mathcal{F}(I)) = \mathcal{F}(\mathcal{P}_{pqkr} \circ (\mathcal{D}_{pqkr}I),
\]

where the apparent sky \( \hat{I} \) is the inverse Fourier transform of \( \mathcal{V}_{pq} \) and \( \mathcal{P}_{pqkr} \) is the inverse Fourier transform of \( \delta_{pqkr} \). Here \( \mathcal{F} \) represents the Fourier transform. Inverting the sum of Eq. 5 over all the baselines results in an estimate of the sky image:

\[
\hat{I} = \sum_{pqkr} W_{pqkr} \mathcal{P}_{pqkr} \circ (\mathcal{D}_{pqkr}I),
\]

where \( W_{pqkr} \) is the weight at the sampled point \( pqkr \). We note that the apparent sky \( \hat{I} \) is now tapered by the baseline-dependent distortion distribution \( \mathcal{D}_{pqkr} \), the latter being the inverse Fourier transform of the baseline-dependent top-hat window:

\[
\mathcal{D}_{pqkr} = \mathcal{F}^{-1}(\Pi_{pq}) = \text{sinc}(\frac{\Delta \Psi}{2}) \text{sinc}(\frac{\Delta \Phi}{2}).
\]

For a source at the sky location \( l \), the \( \Delta \Psi \) and \( \Delta \Phi \) are the phase difference in time and frequency, respectively:

\[
\Delta \Psi = 2\pi \Delta u_{pq}(t, \nu_r)l; \Delta \Phi = 2\pi \Delta u_{pq}(t_k, \nu). \tag{8}
\]

Assuming no other corruption effects apart from decorrelation and assuming naturally weighting a sky with a single source; with decorrelation in effect Eq. 6 becomes:

\[
\hat{I} = \mathcal{P}_{pqkr} \circ (\mathcal{D}_{pqkr}I) = \mathcal{D}_{pqkr}I. \tag{9}
\]

Note that in this formulation, we have assumed that \( \mathcal{P}_{pqkr} = \delta(l) \) is baseline independent as opposed to \( \text{Atemkeng et al. (2020)} \). Eq. 9 is simulated in Figure 1 using the MeerKAT telescope at 1.4 GHz showing the apparent intensity of a 1 Jy source, as seen by the shortest baseline, medium-length baseline, and longest baseline, as a function of distance from the phase center. We see that decorrelation is severe on the longest baseline then followed by the medium length baseline, and that decorrelation is a function of source position in the sky.

### 2. Baseline-Dependent Time and Channel Averaging (BDA)

The distortion distribution, \( \mathcal{D}_{pqkr} \) depends on each baseline and its rotation orientation in the Fourier space which makes the decorrelation to be baseline-dependent. For decorrelation to be baseline-independent, the rectangular sampling bin \( B_{kr}^{[\Delta u_{pq} t, \Delta u_{pq} \nu]} \) across which the data is averaged must be kept baseline-dependent as opposed to the fixed sampling bin currently employed in radio interferometer correlators:

\[
B_{kr}^{[\Delta u_{pq} t, \Delta u_{pq} \nu]} = [t_k - \Delta u_{pq} t/2, t_k + \Delta u_{pq} t/2] \times [\nu_r - \Delta u_{pq} \nu/2, \nu_r + \Delta u_{pq} \nu/2], \tag{10}
\]
where the integration intervals $\Delta u_{pq}^t$ and $\Delta u_{pq}^\nu$ are now also baseline-dependant. In this case Eq. 6 becomes:

$$\tilde{I} = \sum_{pqkr} W_{pqkr} P_{pqkr} \circ (D I),$$

(11)

where $D = D_{pqkr} = D_{\alpha\beta kr}$ is the distortion distribution, which is now equal across all the baselines, $pq$ and $\alpha\beta$ no matter their orientation. We provide details on the implementation of $D$ in Sections 3 and 4.

3. Technologies

The core of Xova’s BDA algorithm is implemented using two recent parallelisation and acceleration frameworks: (1) Dask (Rocklin 2015) is a Python parallel computing library that expresses programs as Computational Graphs whose individual tasks are scheduled on multiple cores or nodes. Dask collections abstract underlying graphs with familiar Array and Dataframe interfaces. (2) Numba (Lam et al. 2015) a JIT compiler that translates the BDA algorithm, expressed as a subset of Python and NumPy code, to accelerated machine code. These are implemented in Xova as follows: dask-mms (Perkins et al. 2021) exposes Measurement Set columns as dask arrays for ingest by Xova then Codex Africanus (Perkins et al. 2021) a Radio Astronomy Algorithms Library, applies BDA, implemented in numba to dask arrays, producing averaged dask arrays and dask-mms writes the averaged dask arrays to a new Measurement Set.

4. Xova

For each baseline (See Figure 2): Measurement Set timeslots are aggregated into averaging bins until $\text{sinc}(\Delta \Psi/2)$ falls below decorrelation tolerance $D$. The acceptable corresponding change in frequency $\Delta \Phi = 2 \text{sinc}^{-1}(D/\text{sinc}(\Delta \Psi/2))$ is calculated and channel width $\Delta \nu$ is derived from $\Delta \Phi$ and used to divide the original band into a new channelisation.
Figure 2. The parts of the baseline closer to the phase centre are subject to greater averaging.

5. Results

Figure 3 shows the image of a high-resolution data set imaged without BDA (right panel) and with 95% decorrelation tolerance BDA (left panel). We note that BDA does not distort the image when compared to the no BDA image.

Figure 3. BDA (left) vs. no BDA (right).

Acknowledgments. The research of Oleg Smirnov is supported by the South African Research Chairs Initiative of the Department of Science and Technology and National Research Foundation.

References

Atemkeng, M., Smirnov, O., Tasse, C., Foster, G., & Makhathini, S. 2020, Monthly Notices of the Royal Astronomical Society, 499, 292
Lam, S. K., Pitrou, A., & Seibert, S. 2015, in Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (New York, NY, USA: Association for Computing Machinery), LLVM ’15. URL https://doi.org/10.1145/2833157.2833162
Perkins, S. J., et al. 2021, in ADASS XXX, edited by J.-E. Ruiz, & F. Pierfederici (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD
Rocklin, M. 2015, in Proceedings of the 14th Python in Science Conference, edited by K. Huff, & J. Bergstra, 130