Radio Astronomy Source Localization with MUSIC and Bluebild Algorithm

Chun-Tso Tsai, Majdouline Ait Yahia, Koloina Randrianavony

***Abstract*—Utilizing antenna array has been a prevailing approach in the field of radio astronomy and telecommunication. In this work, we focus on the application of source localization. That is, to estimate the location of light sources in the sky field. We investigate two approaches using MUSIC [1] and Bluebild algorithm [2]. With data from simulation and real-world observation, we investigate and compare the performance and characteristics of the two algorithms under different parameter settings. By the experiments, we conclude that these 2 algorithms can give us accurate results using fairly small processing time with proper parameter settings.**

I. Introduction

This work is mainly based on the result of [2] and [1]. We follow the light source model defined in [2] to formulate the relationship between the source intensity and the antenna observation. We briefly introduce the model in section II-A, and explain the basic principles of the algorithms. One can find more details in the original papers.

We provide the experiments of the MUSIC and Bluebild algorithm in section III. We discuss the different parameter settings including the threshold, photograph exposure time, etc. Meanwhile, to test the effect of light source settings, we use the simulated data instead of the real-world data. We can also verify the correctness of the light measuring model by showing that the algorithms also work with simulated data.

II. Methods

We follow the model proposed in [2]. The details are stated below.

A. Model of Light Measuring

We assume the light sources emit the source signal as S.

$$S(r) = \hat{s}(r)e^{j2\pi f_c t}, \qquad \hat{s}(r) \sim \mathcal{N}_c(0, I(r)) \tag{1}$$

where $r \in \mathbb{S}^2$ is the position of light source on the sky ; f_c is the frequency of the signal; $\mathcal{N}_c$ is the complex Gaussian random variable; and I is the source intensity.

The model also assumes the light coming from different direction is uncorrelated. i.e.

$$\mathbb{E}\left[S(r_1)S^*(r_2)\right] = 0, \forall r_1 \neq r_2 \in \mathbb{S}^2 \tag{2}$$

Assume we have L antennas at locations $p_1, \ldots, p_L \in \mathbb{R}^3$. The signal received at the ℓ-th antenna is defined as

$$X_\ell = \int_{\mathbb{S}^2} S(r)a_\ell^*(r)e^{-j\frac{2\pi}{\lambda_c}\langle p_\ell, r\rangle}dr \tag{3}$$

where $a_\ell : \mathbb{S}^2 \mapsto \mathbb{C}$ is the sensitivity of antenna; λ_c is the wavelength of the signal; and $e^{-j\frac{2\pi}{\lambda_c}\langle p_\ell, r\rangle}$ is the phase delay due to the position difference between the antennas.

Let's denote the signals received by all antennas as $X = [X_1, \ldots, X_L]^\top$. Usually, due to the large amount of raw data (high data rate and large amount of antenna), there will be a hardware implementing beamforming to compress the data. Let the beamforming weight be $W \in \mathbb{C}^{L\times M}$. The compressed data is then

$$Y = W^H X \in \mathbb{C}^M \tag{4}$$

where M is the dimension of the compressed data.

Given the following model, our goal is to reconstruct the source light intensity $I(r)$ based on the correlation of our observation $\Sigma = \mathbb{E}\left[YY^H\right]$.

B. Signal Estimation using MUSIC Algorithm

We use the MUSIC to estimate the Direction of Arrival (DOA). As staed in the work of [1], the spatial difference of the antennas causes phase difference when receiving signals. Since MUSIC algorithm originally is designed to estimate the frequencies of sinusoidal signals, we can estimate DOA by treating the phase difference as the sinusoidal basis. Observe equation (3), the phase difference between antennas is determined by the inner product $\langle p_\ell, r\rangle$. Because MUSIC algorithm can estimate the inner product term, knowing the antenna placement p_ℓ allows us to estimate the signal location r.

More precisely, given the correlation $\Sigma = \mathbb{E}\left[YY^H\right]$ and the beamforming matrix W. In discrete setting, let $\hat{r} = \left[\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_{n_{px}}\right]^\top \in \mathbb{R}^{n_{px}\times 3}$, where each row $\hat{r}_i$ corresponds to the Cartesian coordinate of the sampled pixel in the sky with totally n_{px} pixels. Assume $a_\ell(r) = 1\ \forall r, \ell$, we denote $A \in \mathbb{R}^{L\times n_{px}}$ to be the phase difference matrix where

$$A_{ij} = e^{-j\frac{2\pi}{\lambda_c}\langle p_i, \hat{r}_j\rangle} \tag{5}$$

So we can apply MUSIC based on the following equation to estimate the variance of source signal S (assume S is already discretized in $\mathbb{R}^{n_{px}}$), which corresponds to the intensity of the source signal.

$$\Sigma = \mathbb{E}\left[YY^H\right] = W^H A\, \mathbb{E}\left[SS^H\right] A^H W \tag{6}$$

C. Signal Estimation using Bluebild Algorithm

Bluebild expresses the estimation of the covariance matrix $\kappa_S = \mathrm{Cov}(S)$ in a compact orthogonal basis B using a functional PCA decomposition which resolves the drawback of computing both κ_S and the intensity function I_S which can

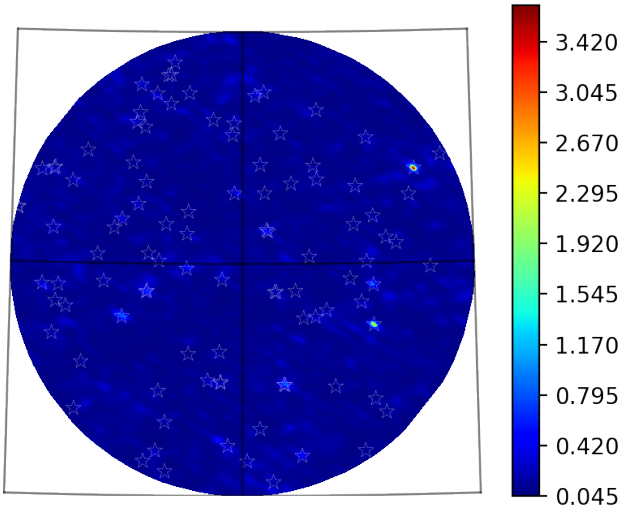

Fig. 1: The DOA estimation using MUSIC.

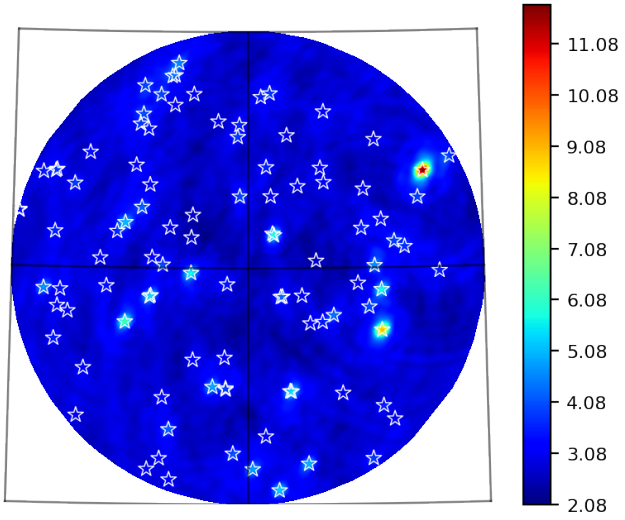

Fig. 2: Comparison of detected intensity with real star positions. (Exposure time = 8 hours; threshold $\tau = 0.8$)

be significant for large instruments. Moreover, we don't need to compute the inverse of the Gram matrix that can be costly if the basis functions are strongly correlated. In addition to the numerical stability and computational efficiency of Bluebild, computing both functions can be accurate and provide efficiency trade-off. Also, by decomposing I_S into distinct energy-levels, it allows one to tailor the post-processing to each energy-level to obtain better estimates.

III. Result

A. Test of MUSIC

We implemented the MUSIC algorithm for direction of arrival (DOA) estimation. The estimated intensity is shown in Figure 1, where the star signs represent the ground truth of the real locations of stars. We can see that MUSIC only captures a few brightest spots, but the estimated location is quite sharp and precise. Also, MUSIC requires only 1 picture to compute the result, and it computes it very quickly (within 1 second). Even if we average multiple pictures over a long time duration, the result remains the same.

B. Test of Bluebild on Real Data

We implemented the Bluebild algorithm following algorithm 2 and 4 in [2], which is based one the model described in Section II.

Fig 2 shows the estimated intensity. We chose the exposure time to be 8 hours because it's the maximum amount of data we have; and the threshold $\tau = 0.8$ is a hand-tuned value to have a clean picture. We can observe the brightest 2 stars on the right is also captured by the algorithm as using MUSIC. When we compare Bluebild to MUSIC, the background noise is larger but we find more light sources. Second, the signal "spreads out" more in the estimation of Bluebild than in MUSIC. Unlike in MUSIC it's a sharp source point, it's a smooth bell-shape pulse over the light source in Bluebild. Thrid, it's also worth mention that to process each picture (each snapshot), the speed of Bluebild is approximately 5 times quicker than MUSIC. It shows that Bluebild not only is more sensitive to the light sources, it is also computationally more efficient.

We can also boost the intensity of other darker sources by equalizing the all the eigenvalues. Namely, if we assume all eigenvalues to be 1, we obtain the result of Figure **??**. We can see that after equalizing the eigenvalues, we are able to locate more light sources. It's worth noting that other than getting larger background noise, we also lose the relative intensity between the stars if we apply such method.

Observe that the user can adjust 2 hyper-parameters for the intensity estimation: the *threshold* of eigenpair selection and the *exposure time*. We first observe the difference of choosing the eigenpair threshold. This is equivalent to the problem of choosing K largest eigenvalues in PCA. In this algorithm, we utilize a threshold τ to choose K eigenpairs as the smallest

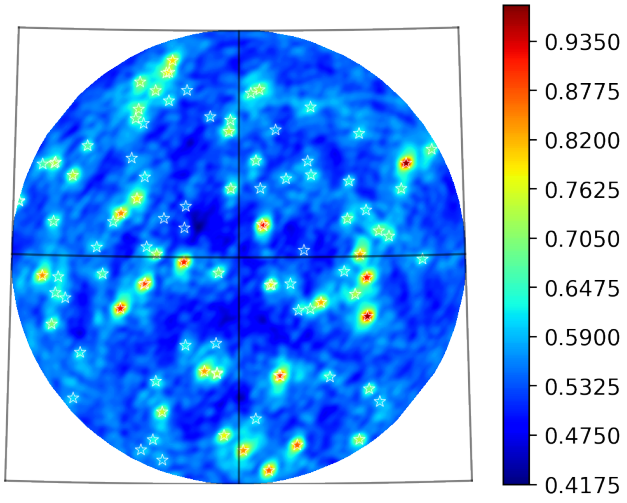

Fig. 3: Estimation after eigenvalues equalized. (Exposure time = 8 hours; threshold $\tau = 0.8$)

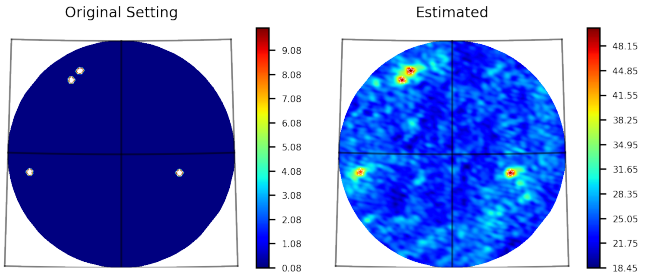

Fig. 4: Generated intensity and the estimated intensity

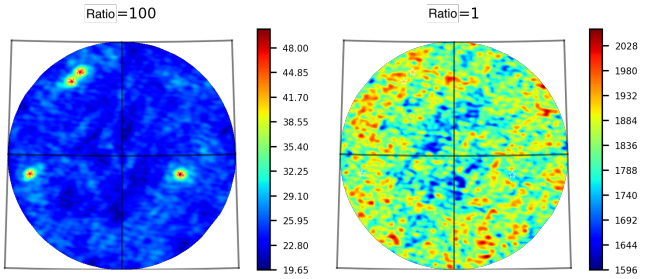

Fig. 5: Comparison under different intensity ratio.

amount of eigenvalues whose summation exceeds τ. i.e.

$$K = \min_{K' \in \mathbb{N}} \left\{ K' : \tau \leq \sum_{k=1}^{K'} \lambda_k / \sum_{m=1}^{M} \lambda_m \right\} \tag{7}$$

The comparison between different threshold values is provided in Figure 6. We may observe that with low threshold, there's less noise but only the brightest signals can be capture. This is useful when we only need to locate the brightest star. When we increase the threshold, more light sources can be identified as the hot spots of the intensity function. But at the same time, the background noise is larger.

Second, we investigate the effect of different exposure time. The result is shown in Figure 7. When we only take a few snapshots, the result contains a lot of noise. On the contrary, when we aggregate over several hours of observation, the noise level dropped to an acceptable value. This is because that with more realization, the variance of the estimation gets lower as well. Figure 7 uses the normal eigenvalues, and we show the result with equalized eigenvalues in Figure 8. We see that it requires more realizations to mitigate the noise, and it's more sensitive to the noise.

C. Simulation Data Generation

To further verify the accuracy of the sky measuring model and the algorithm, we follow the model stated in Section II-A to generate our own data. We generate the data using different ratio between the intensity of light sources and background noise. Figure 4 shows the result of using ratio $I_{src}/I_{bg} = 100$, with 53 minutes of time exposure. We can see that the location of the stars are successfully identified. Yet, the intensity values are different among the original one and the estimated one. Thus we can confirm that the Bluebild algorithm can only estimate the distribution of the intensity function but not the exact intensity of the light source.

Given that we have the ability to generate our own data, we can now test various situations. Figure 5 shows 2 experiments with the intensity ratio by 100 and 1. As expected, the estimated signal looks like pure Gaussian noise when the intensity of source and background noise has similar level.

IV. Conclusion

We show that the light measurement model stated in section II-A is close to the real situation by our experiment. Both MUSIC and Bluebild algorithm provide efficient solution to compute the light intensity based on beam-formed radio data, and they can both capture a few brightest sources with ease. Yet, it's more difficult for MUSIC to capture darker sources. However, if we only need the brightest sources, MUSIC can obtain the result with less noise and sharper response.

On the other hand, the Bluebild algorithm is able to achieve a satisfactory result when given enough data. One can utilize our conclusion towards the choice of hyper-parameters to choose the best configuration for their applications.

Although we focus on the source localization in this report, these algorithms can also apply to other applications that involves processing beam-formed data. As MUSIC can analyze audio signals as well as images, these methods are flexible for various applications. Meanwhile, to improve the performance, one can modify the algorithm based on their application.

Acknowledge

The authors thank Prof. Andrea Ridolfi for the lectures to equip us with the essential knowledge to work on this topic. We thank our assistant Sepand Kashani for the excellent work of the paper, as well as the kind assistance and clear explanation for the authors.

References

[1] H. Tang, "Doa estimation based on music algorithm," p. 56, 2014.
[2] S. Kashani, "Towards real-time high-resolution interferometric imaging with bluebild," 2017. [Online]. Available: http://infoscience.epfl.ch/record/269252

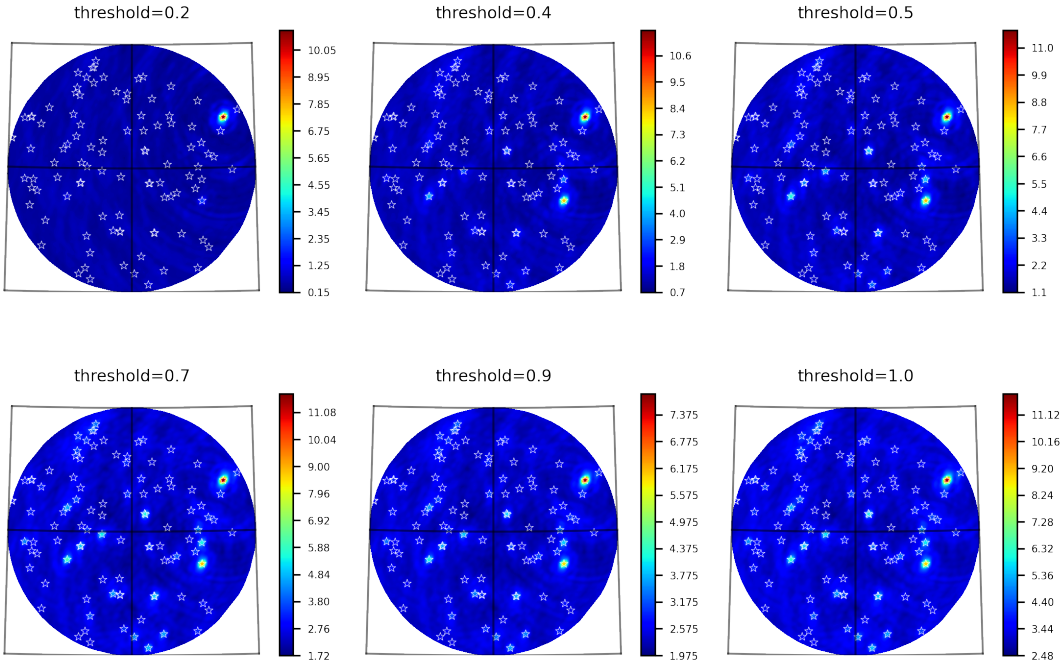

Fig. 6: Intensity given different thresholds. (Exposure time = 8 hours)

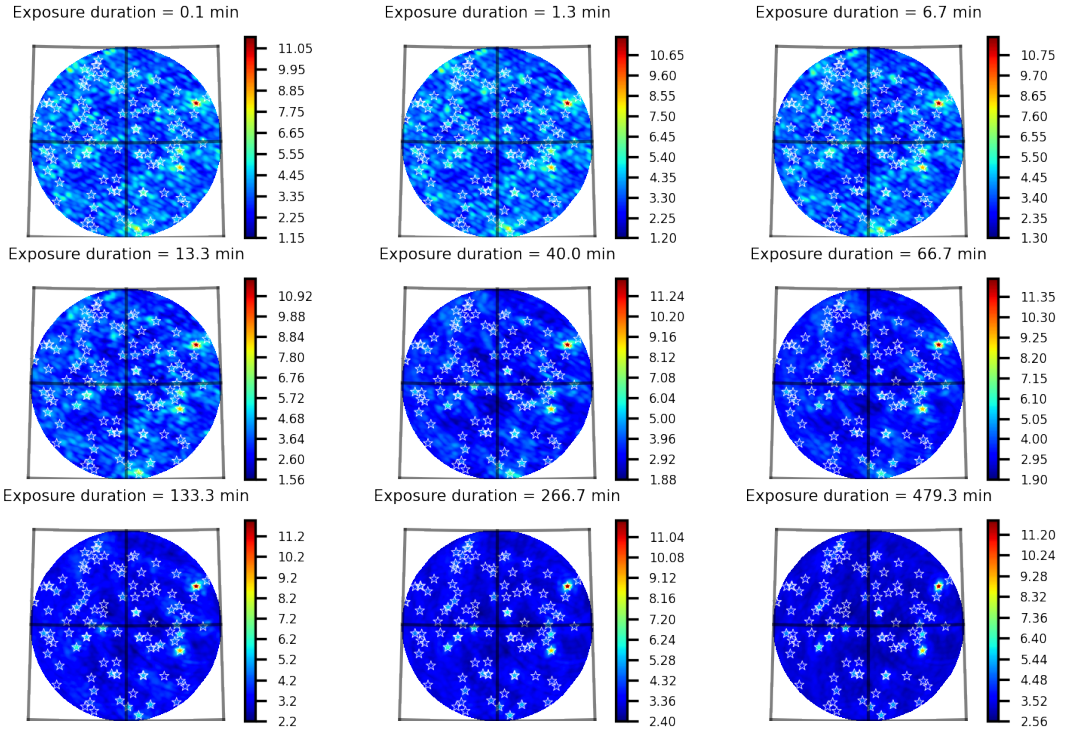

Fig. 7: Intensity given different exposure time with normal eigenvalues.

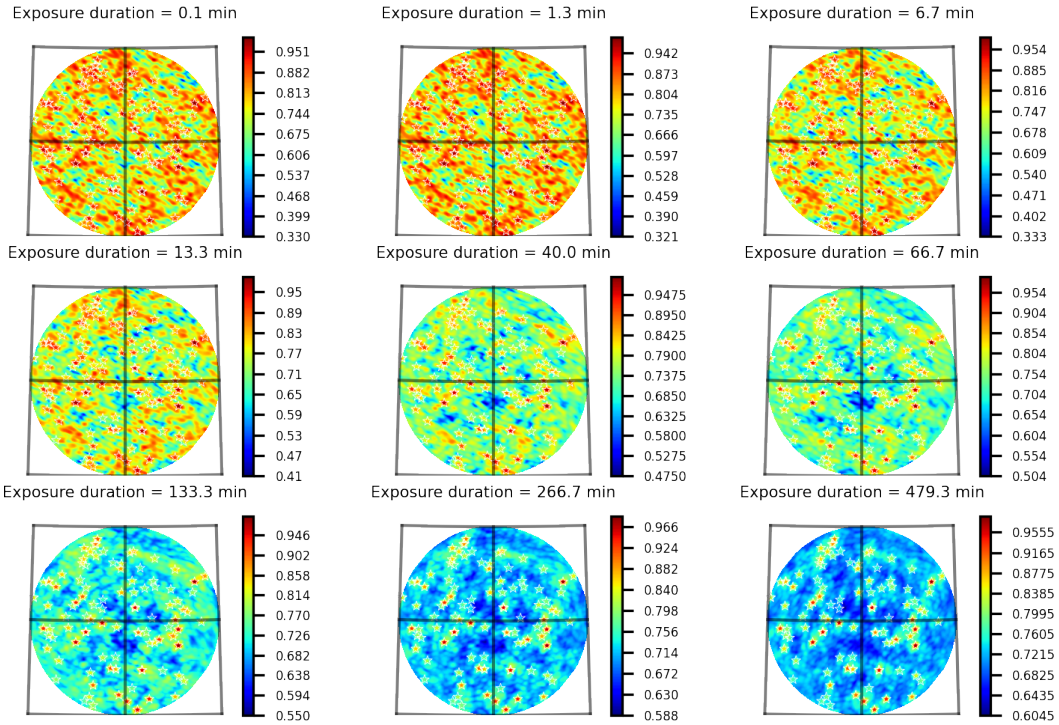

Fig. 8: Intensity given different exposure time with equalized eigenvalues.