DOI: 10.1111/cgf.70117
Eurographics Conference on Visualization (EuroVis) 2025
W. Aigner, N. Andrienko, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 3

Lactea: Web-Based Spectrum-Preserving Multi-Resolution Visualization of the GAIA Star Catalog

Reem Alghamdi[1], Markus Hadwiger[1], Guido Reina[2], and Alberto Jaspe-Villanueva[1]

[1]King Abdullah University of Science and Technology (KAUST), Saudi Arabia
[2]Visualization Research Center (VISUS), University of Stuttgart, Germany

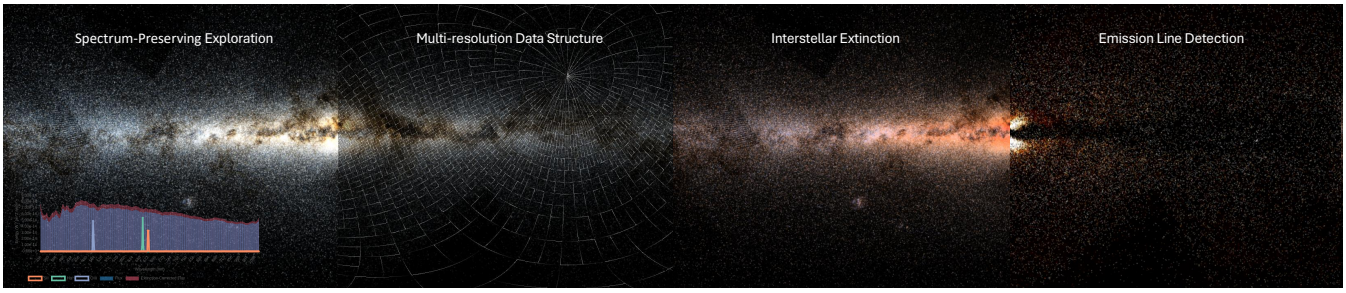


Figure 1: *Panoramic view of the Milky Way by* Lactea *tool, showing different features and visualization of the presented spectral rendering algorithm.*

Abstract

The explosion of data in astronomy has resulted in an era of unprecedented opportunities for discovery. The GAIA mission's catalog, containing a large number of light sources (mostly stars) with several parameters such as sky position and proper motion, is playing a significant role in advancing astronomy research and has been crucial in various scientific breakthroughs over the past decade. In its current release, more than 200 million stars contain a calibrated continuous spectrum, which is essential for characterizing astronomical information such as effective temperature and surface gravity, and enabling complex tasks like interstellar extinction detection and narrow-band filtering. Even though numerous studies have been conducted to visualize and analyze the data in the SciVis and AstroVis communities, no work has attempted to leverage spectral information for visualization in real-time. Interactive exploration of such complex, massive data presents several challenges for visualization. This paper introduces a novel multi-resolution, spectrum-preserving data structure and a progressive, real-time visualization algorithm to handle the sheer volume of the data efficiently, enabling interactive visualization and exploration of the whole catalog's spectra. We show the efficiency of our method with our open-source, interactive, web-based tool for exploring the GAIA catalog, and discuss astronomically relevant use cases of our system.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization;*** • ***Computing methodologies*** → ***Rendering;*** • ***Applied computing*** → *Astronomy;*

1 Introduction

Astronomy is recently undergoing a remarkable information flood of both simulated and observed data due to advances in computational infrastructures. For instance, the spacecraft of the GAIA mission, which was launched a decade ago by the European Space Agency (ESA) [tea16], revolutionized the field by creating an exact three-dimensional map of almost two billion stars from the Milky Way and beyond, playing a pivotal role in recent astronomical breakthroughs [Eur23, CCM*20, GAM*18]. GAIA data release 3 (DR3), published in June 2022 [tea23], contains a large variety of astronomically significant attributes and error measurements. This third data catalog includes 1.8 billion objects, with 1.46 billion of these radiation sources (most of them stars) containing the sky position (right ascension and declination), parallax (distance), and proper motion (velocities). Additional parameters, such as the photometry, are also recorded. Furthermore, low-resolution calibrated continuous spectra, called *XP spectra* or *BP/RP spectra*, have been inferred for blue-pass and red-pass sensors for a little over 200 million stars (see Sec. 3.2). Through the analysis of the light spectrum emitted

by a celestial object, comprehensive star characterizations, such as estimated effective temperature, surface gravity, and the chemical components, are derived [A*23]. Astrophysics visualization has emerged as an interdisciplinary field bridging astronomers and visualization experts to develop techniques enabling scalable exploration and understanding of astrophysical data, fostering scientific discovery, methodological validation, and novel insights about the universe [LYA*21]. From a visual computing perspective, astronomical datasets present unique challenges and opportunities due to their high dynamic range, extensive memory and computational demands, and the need for algorithms that scale with data complexity. Effective visual representations often prioritize conveying insights over physical accuracy, e.g., the mapping of stellar spectra to color—a challenging problem—can be further complicated by conventional techniques, such as linear color maps, which may obscure critical details or fail to emphasize meaningful patterns [HH21].

Due to those challenges and GAIA's significance in astronomy, previous work has proposed techniques for exploring the GAIA data (see Sec. 2). However, the spectral information available in the data was not explored previously. Our research aims to take the first step into real-time spectral exploration by using novel visualization techniques. We consider the light spectra of hundreds of millions of stars in real-time, enabling an interactive visualization in various projections, color mapping, and spectral manipulation schemes to explore the entire dataset (around 630 gigabytes in size). **The main contributions of this paper are**:

- A novel multi-resolution, spectrum-preserving data structure for two-dimensional distributions of incoming light, designed to allow interactive visualization with arbitrary data sizes.
- A progressive rendering algorithm for massive, real-time light spectrum exploration.
- Three visualization use cases to show the application of these techniques in real scenarios.
- An open-source[†], interactive, web-based prototype that applies these techniques to the GAIA catalog, making this the first work, to the best of our knowledge, to manage the GAIA spectra for real-time exploration and visualization.

2 Related work

Astrophysics visualization. Astronomy is experiencing an explosion of simulated and observed data, fueling the need for advanced visualization approaches, which also drive progress in the visualization field. Lan et al. [LYA*21] summarize a decade of astrophysics visualization, categorizing work by task (e.g., feature identification [KHA12]), technique (e.g., 3D rendering [SJMS19]), topic (e.g., galactic astronomy [VE20]), and data source (simulated [AMJV*23] or real [BAC*20]). Following that scheme, our technique supports data exploration and feature identification with interactive 2D images and plots of observed galactic data.

Real time space visualization. Various work has been done on the real-time visualization of the universe. OpenSpace by Bock et al. [BAC*20] is one essential system for exploring extensive astronomy datasets, suitable for educational and scientific research. They use a dynamic scene graph [ACS*17] to manage the different scales of the data. It stands out for its ability to integrate with existing tools and catalogs and to create cinematic scenes of the universe in 3D. While the GAIA data is one of the catalogs OpenSpace is capable of handling, the spectral distribution is not utilized for exploration or visualization. Another system is Gaia Sky by Sagristà et al. [SJMS19]. It is an open-source tool for exploring our galaxy in 3D, using data from the Gaia Catalog. It supports real-time navigation among millions of stars, featuring advanced simulations like relativistic effects and gravitational waves. Its main strength lies in its ability to provide detailed, interactive celestial tours with high efficiency and precision. The color information is computed using temperature, inferred from the spectrometers of the GAIA telescope. It does not deal with the source spectral information.

GAIA visualization. In addition to Gaia Sky and OpenSpace, other studies have addressed visualizing GAIA data with InfoVis and SciVis approaches. Moitinho et al. [M*17] created the Gaia archive Visualization Service as a web-based InfoVis tool that allows for interactive exploration of Gaia and TGAS data through customizable histograms and scatter plots. Although scientists are the target, no spectra visualization is provided. In contrast, The Mikulski Archive for Space Telescopes (MAST) project [MHL*18] provides various exploration tools for multiple data archives, including GAIA. Their system supports spectral viewing as an InfoVis tool. Deborah et al. [BGR*16] introduced ESASky, a web-based data exploration and visualization tool. It works with a subset of multiple astronomy data catalogs, including GAIA. However, it uses conventional transfer functions for a single attribute simultaneously. Lastly, high-performance visualization tools can produce heatmaps of the GAIA dataset, such as vaex [BV18] or mosaic [HM24]. We note that no work has been done to visualize and explore the spectra of the GAIA catalog before our work.

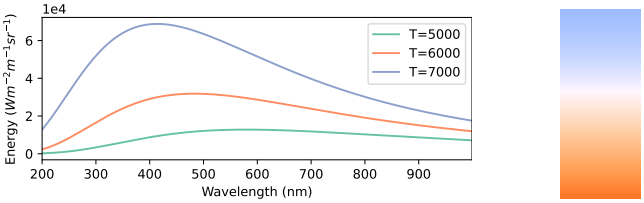
Light phenomena visualization. Regarding the light spectra, simulating light phenomena in diverse astronomical contexts by manipulating the light is possible thanks to the advancement of GPUs. For instance, Verbraeck and Eisemann [VE21] simulate the light bending in the proximity of a black hole, while Consta et al. [CBE*20] simulate the light in the atmosphere of different planets. The Doppler effect due to the high velocity of moving galaxies has been simulated as well [AMJV*23].

Massive point-based rendering. GAIA's vast dataset requires point cloud methods, widely used in visualization for applications like LiDAR and photogrammetry [GP11]. Gobbetti and Marton's Layered Point Clouds [GM04], an LoD-based octree structure, efficiently divides datasets into nodes that collectively recreate the original set, enabling high-quality rendering focused on regions of interest. This approach has influenced modern web-based 3D point cloud visualizations [RGM*12], such as Schütz et al.'s Potree [S*16], which leverages out-of-core techniques for interactive rendering of billions of points, achieving substantial performance gains [SOW20]. However, while these methods handle scalar attributes or simple colors, our work addresses stars with multiple attributes and light spectrum histograms, requiring several kilobytes per point.

3 Background

This section provides the necessary foundation to understand the context of our work. First, key concepts on light, spectra, and relevant astronomical applications are discussed. Then, the GAIA dataset, which is used to evaluate our algorithm, is outlined.

[†] https://github.com/vccvisualization/lactea



(a) (b)

Figure 2: *(a) Blackbody radiation as expressed by Planck's law at different temperatures. (b) The normalized CIE color of blackbody radiation from* $3000K$ *to* $10,000K$*. Cooler stars appear redder, while hotter stars are blue.*

3.1 Physics of Light

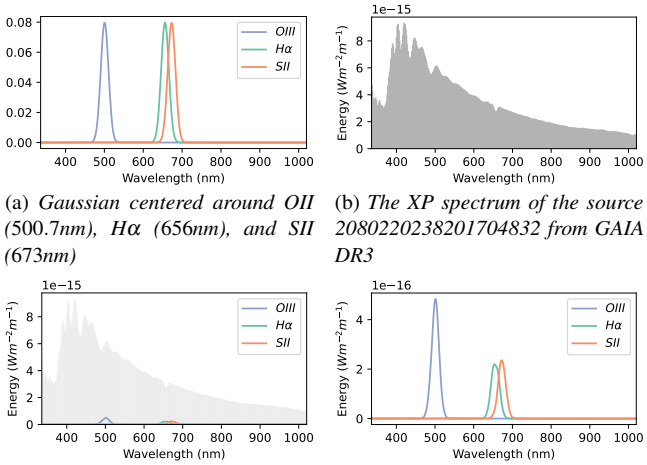
Light is emitted by matter in numerous ways. In the case of stars, light is emitted based on their surface temperature and the chemical components of their surface [CO17]. The continuous star radiation can be modeled as an idealized black body, which is a theoretical physical body that absorbs all radiation. The spectral distribution is calculated by Planck's law as given in Eq. 1:

$$B_\lambda(T,\lambda) = \frac{2hc^2}{\lambda^5}\left[\exp\left(\frac{hc}{\lambda k_B T}\right) - 1\right]^{-1}. \tag{1}$$

Here, λ is the wavelength, h is Planck's constant, k_B is Boltzmann's constant, c is the speed of light, and T is the temperature [K*17].

The black-body radiation is determined by the temperature alone, not by the star's size or elemental composition. Fig. 2a shows Planck's law distributions at different temperatures. The second factor that contributes to the emission spectrum is related to the chemical nature of a star's surface. When an atom of a chemical element absorbs a photon with the exact energy required for an electron to make an upward transition from a lower to a higher orbital, absorption lines in the continuous stellar spectra are observed. In contrast, emission lines occur when an electron transitions downwards from a higher to a lower orbital [CO17].

The chemical elements found in a star are identified by matching both types of lines against an observed spectrum. Scientists have identified the spectral lines of many elements [RKR*97], such as hydrogen's Balmer lines. Sometimes, looking at only part of the spectra for analysis is desirable to find features of interest, especially when the rest of the spectrum overshadows such features. For instance, isolating the spectra generated by the first hydrogen Balmer line, called hydrogen alpha $H_\alpha = 656nm$ in the red region of the spectrum. This can be useful in many astronomical contexts, such as the spectral classification of nebulae [Hea90]. This type of imaging is called narrow-band filtering. In the real world, dedicated lenses are used to obtain the gray-scale image that shows the intensity of the filtered wavelengths. It is possible to obtain a narrow-band filtered image from a spectral image computationally. Fig. 3 shows how narrow-band filtering works. The transmission function is, for example, modeled as a Gaussian centered around the element of interest with a specified width. Then, the filter multiplies the spectrum, and the result is integrated to obtain the final intensity [BD07]. This process is formulated in Eq. 2, where I is the intensity or flux received through the narrow filter, $f(\lambda)$ is the light spectrum, and $T(\lambda)$ is the transmission function:



(a) *Gaussian centered around OII (500.7nm), Hα (656nm), and SII (673nm)*

(b) *The XP spectrum of the source 2080220238201704832 from GAIA DR3*

(c) *The star spectrum, and the result of convolving with the Gaussians drawn on top*

(d) *Resulting Integrated Spectra*

Figure 3: *A star spectrum is integrated with three different filters.*

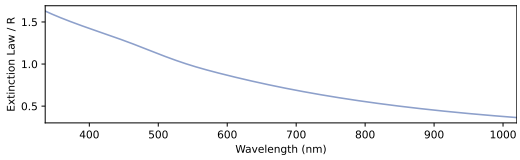


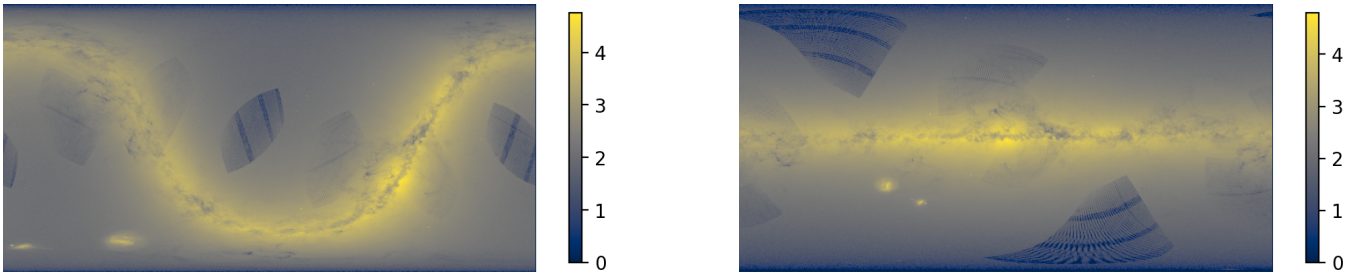
Figure 4: *The Fitzpatrick [Fit99] normalized extinction curve. The law corrects blue radiation more than red because extinction is wavelength-dependent: shorter wavelengths (blue) are scattered more easily than the longer ones (red).*

$$I = \int f(\lambda)T(\lambda)\,d\lambda. \tag{2}$$

Many other elements can also be isolated, such as sulfur-II $673nm$ or oxygen-III $OIII = 500.7nm$. The famous Hubble palette imaging [Gre04] is obtained by filtering by these three elements, then mapping the resulting intensity into $RGB = (I_{SII}, I_{H_\alpha}, I_{OIII})$ for the classical SHO Hubble palette or $RGB = (I_{H_\alpha}, I_{OIII}, I_{OIII})$ for the bi-colored HOO image or any of their combinations. In the real world, various factors influence how light is perceived, such as the Doppler shift due to the distribution of velocities in a gas or changes in the pressure. Another phenomenon that can change how light is perceived is *interstellar extinction* [CO17]. It occurs when dust clouds populating the Milky Way obscure or change how we perceive electromagnetic radiation due to the intervening dust scattering and absorbing starlight. Extinction is wavelength-dependent; red light is not as strongly scattered as the shorter blue light. Hence, a *reddening* effect is observed in the spectra, leading stars to appear redder than their effective temperature implies. Extinction detection is essential to reconstruct the original spectra and leverage the spectral data effectively [DBRD*18]. In a sense, extinction can be thought of as the reduction of the intensity as a result of light scattering. If $f(\lambda)$ is the observed XP spectra, the corrected $F(\lambda)$ is computed by

$$F(\lambda) = f(\lambda).10^{0.4A_0A'(\lambda)/R}. \tag{3}$$

Here, A_0 is the monochromatic extinction [HK82] given by GAIA, and $A'(\lambda)$ is the wavelength-dependent extinction curve, and R is an extinction parameter set to 3.1. This paper uses the



(a) *Equirectangular projection.* (b) *Galactic coordinates use the sun as the origin.*

Figure 5: *The log-density map of sources with XP spectra available from the GAIA catalog in equirectangular projection (a) and in galactic coordinates (b) . The leaf-shaped empty regions are due to the corresponding data being excluded from the GAIA release.*

Fitzpatrick extinction law [Fit99] following the GAIA documentation [gai24b]. However, other curves specifically made for GAIA via further analysis and deep learning methods can be found in the literature [BLWW20, ZYH*24]. Fig. 4 shows the extinction law (normalized by R=3.1) as a function of the wavelength.

In astronomy, mapping a star spectrum to a color is an open research problem without a single astrophysically motivated standard color palette [HH21]. However, a common way of obtaining a color is by convolving the light spectral distribution with the three sensitivity functions based on the human perception model to obtain CIE XYZ values. Then, XYZ values are linearly transformed to an RGB colorspace and normalized [PJH16]. Using this coloring method, black bodies defined by Plank's law will have a color between red (cooler) and blue (hotter), as shown in Fig. 2b. Another solution is the spectrum-based RGB color transfer function [AFK*14]. A transfer function is defined over the spectrum distribution. The RGB color is then determined by accumulating the combination of the light flux at a given point and the color from the transfer function at that point [AFK*14]:

$$C = \sum_{i=1}^{n} S(i) * T(i). \tag{4}$$

Here, n denotes the number of bins used to represent the spectrum, C is the resulting RGB color, $S(i)$ is the light flux at bin i, and $T(i)$ represents the color defined in the transfer function at position i. Various other coloring models have been used over the years, such as the B-V color index [Zom06]. However, these methods often result in a loss of the star intensity information at the expense of accurate wavelength-to-color mapping. To this end, it is possible to apply color mapping based on the star's intensity or magnitude. In addition, image processing techniques such as Gaussian smoothing or tone mapping are often applied to highlight features in the image.

3.2 The GAIA Dataset

GAIA data release 3 (DR3) [tea23] contains a large variety of astronomically relevant attributes and error measurements such as sky position, distances, and proper motions. Sky positions are given in the Celestial equivalent of latitude and longitude [Bal13], referred to as *Declination* (DEC) (from -90° to 90°) and *Right Ascension* (RA) (from 0° to 360°) respectively. In addition, photometric and spectrophotometric data were released, allowing astronomers to determine numerous astrophysical information about the observed sources [CWJ*21]. Photometry is a system used to measure the propagation of light waves by using a set of passbands or optical filters with known radiation sensitivity [HK82]. Through a complex calibration process, a set of passbands are defined: G, G_{BP}, G_{RP} and G_{RVS} [JGC*10]. The XP spectra obtained from BP/RP photometry are analyzed to determine the effective temperature, surface gravity, metallicity [M/H], radius, and other attributes [A*23]. Similarly, the RV spectra (not used in this work) are used to estimate the stellar atmospheric parameters, individual chemical abundances, and Diffuse Interstellar Band (DIB) parameters [RB*23]. In the current release (DR3), a subset of 219,197,643 sources were carefully chosen after a thorough study [A*23] to have their continuous BP/RP spectra released publicly [DA*23]. This leads to leaf-shaped artifacts in the resulting image due to the unavailability of stars in those regions. Fig. 5 shows the log-density map of the sources with available XP spectra in two projections: the equirectangular projection (Fig. 5a) and the galactic coordinates (Fig. 5b). The spectra are represented as Hermite polynomials [ZGR23]. Weiler et al. [WCFJ23] analyze and discuss how to use the continuous spectra efficiently to perform many operations and tasks. In addition to the continuous spectra, GAIA released a sampled version of around 34 million stars. Here, the histogram consists of 343 bins from 336nm to 1020nm, sampled each 2nm, measured in the standard physical units $Wm^{-2}nm^{-1}$. Once a passband p_i is available, zero points z_i converting internal fluxes to absolute magnitudes can be derived [CV18]. Then, any flux can be converted to absolute magnitude $m = -2.5log(p_i) - z_i$.

4 Methodology

Our approach consists of three main components: 1) A spectral-preserving, multi-resolution data structure; 2) Progressive, spectral accumulation; and 3) Composition of the final image via filtering, processing, convolving, and rendering to the output view. With this methodology, we generate an interactive visualization and exploration of the dataset for astronomical tasks. Fig. 6 gives an overview of our system.

The data structure is a multi-resolution, binary-tree-based hierarchy of precomputed nodes with light sources information, using 2D polar coordinates over the earth dome. Each node consists of a dual representation of the GAIA objects within that region of the sky: a high-resolution representation of the brightest remaining stars (similar to the Layered Point Cloud structure), and a low-resolution aggregation of the total energy of the lower levels of the hierarchy. The data structure is designed with two aims: to fill the highest levels of detail (LoDs) with the brightest stars and to preserve the energy sum at any LoD. This design choice results in an image that approximates the brightness of the whole data in the lower-level nodes. In addition, and given that all information is stored as discretized spectra, we can construct a spectral image that accurately represents

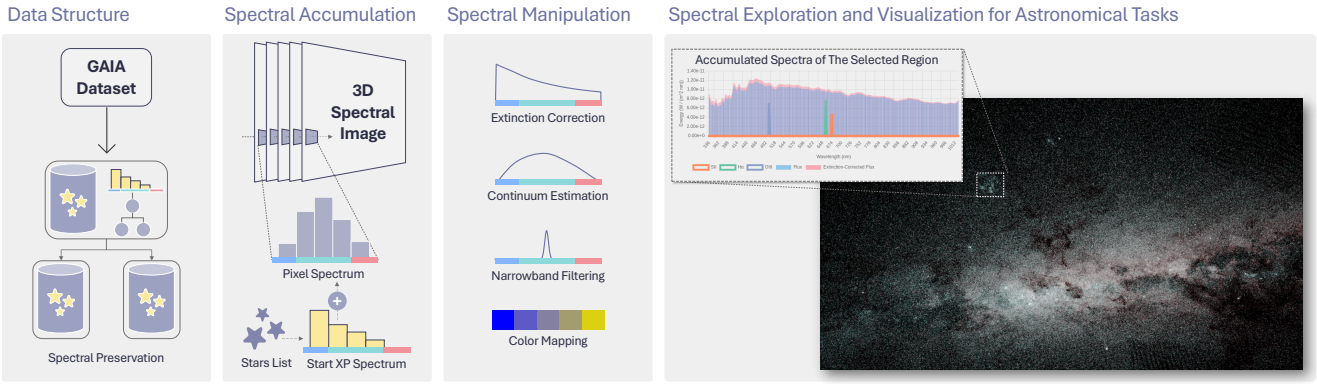


Figure 6: *Overview of the main components of our technique: Using our pre-computed, multi-resolution data structure, we generate interactive views of the galaxy with different projections by accumulating spectral histograms into a GPU 3D buffer in real time, and manipulating them according to different parameters (color mapping, filtering, etc.) to create visualizations and insights that help domain experts to better understand the GAIA data.*

the whole GAIA dataset without traversing the entire tree. The data structure is described in more detail in Sec. 4.1.

The progressive accumulation algorithm employs out-of-core techniques for the multi-resolution data. Thanks to the dual node representation and energy conservation, the tree can be traversed and cut based on the viewing parameters, rendering budget, and the LoD of interest. The inner nodes of the cut tree will raster their star list to the screen as points. Leaf nodes at the borders of the graph cut are represented as patches. The spectra of star points and node patches are aggregated into a 3D buffer, representing the spectral image of the data, where the third dimension stores as energy buckets of a predefined spectral interval of the discretized spectra.

Composition of the final frame is carried out for each pixel's spectrum to fulfill astronomical tasks like narrow-band filtering and interstellar extinction. The spectral information is processed, filtered, and convolved to compute the final color based on the user-defined image processing settings and the desired visualization scheme. The accumulation and image compositing are described in Sec. 4.2, and example astronomical use cases are discussed in Sec. 6.

4.1 Data Structure

Our multi-resolution structure is a generalization of Layered Point Clouds (LPC) [GM04] for point data with many additional attributes contributing to the rendered image. We need to define 1) A maximum number of points per node, 2) the importance scalar metric that will decide which points have more priority in the representation, and 3) an aggregation function to sum up the visualized attributes. Our approach generates a hierarchy over the point samples of the dataset, clustering them into spatially coherent sets arranged in a binary tree. As in LPC, the addition of all the sets of points of the hierarchy sums up precisely the entire dataset. The novelty lies not only in the fact that we sort the points by generic importance criteria but that each node additionally contains a coarse representation of the underlying subtree for each region of the space, which allows for preserving lower-level properties of the data when representing a particular level of detail, irrespective of the chosen tree cut.

Figure 7 shows an overview of how our data structure is utilized. We apply this data structure to explore the spectra of the GAIA catalog objects in the following way: we use the *Magnitude* parameter as the importance metric to decide the priority of the objects in the LoD hierarchy. In other words, we established that the stars whose light contributes most of the energy arriving on the Earth have more importance in our visualization. The property we want to preserve is the amount of light by wavelength, which can be aggregated in a spectrum histogram. We also preserve the photometry bands and the extinction-corrected spectra similarly.

The novelty of our approach lies in the dual representation of the spectral data. Each node of the tree represents an area in the dome space, and contains, for one side, the spectra exact information for each one of the stars included in that node, similarly to LPC. At the same time, it also contains both the spectral histogram of its own stars as well as the sum of all the other stars contained in the nodes of the subtree under it, which we call a spectral patch. This allows us to represent a particular area of the dome by mixing pixels as actual light sources, with its exact spectrum, with spectral patches covering areas of the node (see Figure 7). Thus, with an arbitrary tree cut, the resultant spectrum is exact up to the cut (thus, the ones with higher magnitude) and approximated for the subtrees below it. As shown in the next section, this allows different levels of detail depending on the rendering time budget, as well as progressive rendering.

$$\text{Let } N_A \subset N \text{ where } \forall n \in N_A \begin{cases} n.BoundingBox \text{ intersects } A \\ \wedge \\ Surface(n.BB) \geq k * Surface(A) \end{cases}$$

Let $N_c \subset N_A$ the nodes in the border of the traversal cut

$$Spectrum(A) = \sum^{N_A} \sum^{stars} (wl, flux) + \sum^{N_c} subtree\_spectrum. \tag{5}$$

Eq. 5 reflects this design, where A is the area of the dome in which we want to compute the incoming light spectrum; N_A is the subset of nodes of the tree (N) that intersects with that area, k the traversal-stop factor, where a node surface is considered smaller than a single pixel. The subset of nodes N_C are the ones on the tree cut, defined by a loading time budget per frame. Then, the total incoming light spectrum through the area A is the sum of the pairs of wavelength and flux, $(wl, flux)$, of all the stars included in all the nodes above the tree cut (exact), plus the sum of the spectrum patches of the nodes below the tree cut (approximation). With this data structure, the computed incoming light is always approximated with the whole stars dataset, no matter how much the render time

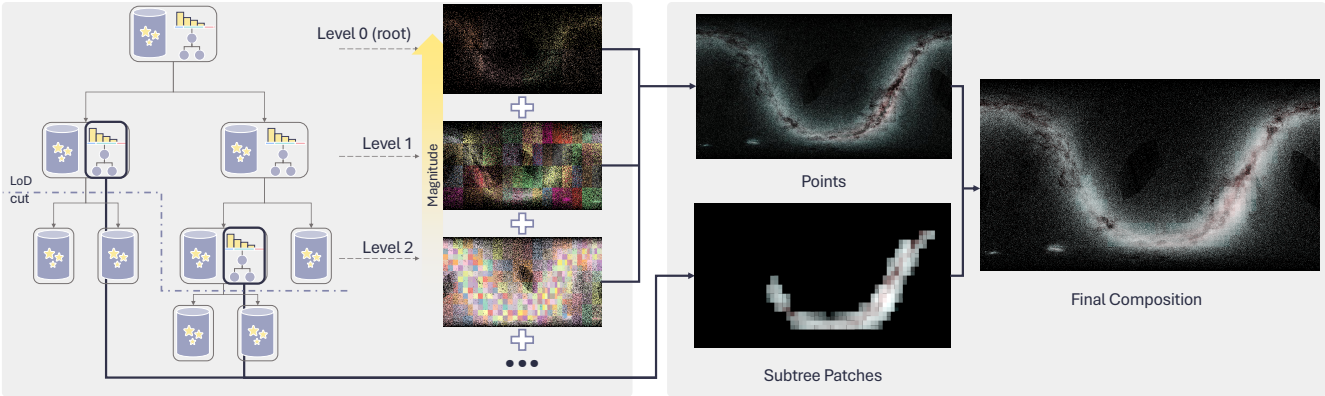


Figure 7: *Our pre-computed data structure partitions the two-dimensional dome space into a binary tree. Each node contains an exact representation of the stars selected according to an importance metric (in this case, brightness) and a coarse representation of the full subtree below. The final composition is a mix between the exact spectra of the stars over the tree LoD cut, and the coarse spectra of the nodes in the cut, enabling a close approximation for all the incremental steps, and converging progressively to the exact spectrum of the incoming light.*

budget is. It also allows to progressively load more and more stars, and the resultant spectrum will converge to the exact solution as we keep loading more stars and fewer spectral patches. Each node stores its ID, bounding box, children ID, clipping coordinate, splitting axis, and the spectra of its own stars and the subtree. In addition, it keeps a list of star IDs belonging to it.

Construction. First, a script is used to download the GAIA CSV files from the official website and convert them into binary files. Those binary files are then merged, and all objects are sorted (in descending order) by their magnitude.

Algorithm 1 Data structure construction

```
▷ —— Top-down pass: tree creation ——
queue.push(root_node)                                          ▷ Whole range
node ← queue.top()
while !queue.empty() do
    node ← queue.pop()
    stars_array.sort(byMagnitude, node.range.min, node.range.max)
    if node.range.length ≥ N then                              ▷ Inner node
        node.stars ← stars_array[node.range.min..N]
        node.computeBoundingBox()
        splitting_axis ← if(node.levelodd) RA else DEC
        stars_array.sort(bySplittingAxis, node.range.min + N, node.range.max)
        queue.push(newnode(LeftRangefromSplittingElement)
        queue.push(newnode(RightRangefromSplittingElement)
    else if node.range.length < N then                         ▷ Leaf node
        node.stars ← stars_array[remainingRange]
        node.computeBoundingBox()
        node.computeSpectrum()                                 ▷ Sum spectra of node's stars
    end if
end while

▷ —— Bottom-up pass: fill course spectra representation ——
for Every level from Bottom to Up do
    for Every Node do
        if node is Inner then
            node.computeSpectrum()
            node.subSpectrum = node.child1.spectrum + node.child2.spectrum
        end if
    end for
end for
```

Algorithm 1 describes the tree-building process. The only input parameter is N, the maximum number of stars in each node, which will be exact for the inner nodes and an upper bound for the leaf nodes. Because of the large amount of data to deal with, the algorithm is designed to work in place on the data, which can be located in main memory or mapped from permanent storage, by moving indices and re-sorting the information. It is composed of two passes: the first one creates the tree nodes and the hierarchy in a top-down fashion (like a k-D tree), by sorting the stars by magnitude and selecting the brightest ones, then sorting the remaining stars again, each time alternating the median split axis orientation for the children, until the number of remaining stars is lower than N. This way, the tree is built in two-dimensional patches along the polar coordinates of the skydome, *DEC* (declination) and *RA* (right ascension). In the leaves, the spectra of the contained stars are aggregated. In the second pass, those spectra are propagated bottom-up to the root, summing up their histograms. Finally, the tree structure is saved to disk. Each node is saved to a separate file containing a maximum of N stars' information and spectrum. Lastly, the subtree spectra of all nodes are stored in a separate file for quicker retrieval.

4.2 Spectral Rendering

At run time, the user explores the dataset by interactively moving the camera along the sky and selecting parameters. Whenever a new interaction occurs, the accumulated spectral frame buffer is recomputed. Then, the data structure is traversed to determine which nodes need to be rendered as stars or as patches. The nodes are loaded from the network through the cache systems, which move the data to the GPU and record their offsets. Then, stars are processed to accumulate their light contribution. Meanwhile, patch IDs are stored in the 3D buffer, and their area is computed to uniformly distribute their spectra over the patch at color computation time. At the color computation stage, each pixel's color is determined by processing the accumulated pixel spectra at the corresponding 3D buffer position, as well as the patch spectrum (if any) divided by its area. An overview of this process can be seen in Figure 8. As the rendering order follows a criterion of maximum contribution, the accuracy of the frame improves progressively in a logarithmic way. Due to the multi-resolution nature of the data structure, the rendering process is done progressively until the star and patch queues are empty. This also allows for a smooth, interactive interface.

Creation of the rendering queues. When the camera moves, the view bounding box is computed based on the projection type and the camera parameters. Then, the tree is traversed, and the star and spectral patch queues are constructed, until the cut-off criterion is met. If a node is at the border of the tree cut, the node is added to another queue, called the patch-rendering queue, and the recursive traversal stops for that subtree. Both queues are priority queues - brighter nodes are loaded first. As explained in Sec. 4.1, the node's

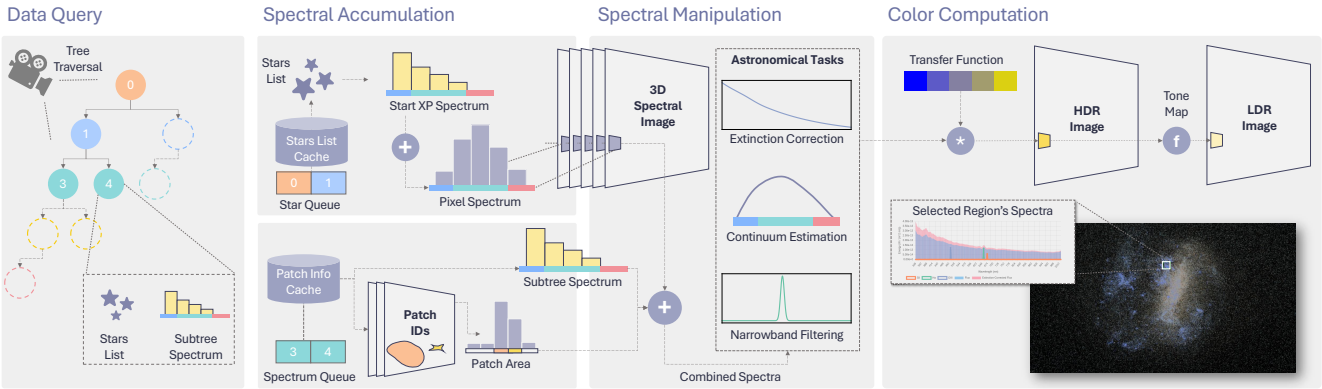


Figure 8: *Overview of the rendering process: Querying the data according to the camera position, spectral accumulation in the GPU 3D buffer, spectral manipulation with user-defined parameters, and color convolution per pixel.*

double representation of the GAIA sources allows for accurate spectral representation without the need to traverse the whole tree (at the cost of spatial resolution). Stars are rasterized as individual points in space. The intensities of the subtree spectrum will be uniformly distributed across the area covered by the corresponding patch.

Traversal cut-off strategy. A common spatial multiresolution cut-off strategy is based on the screen-space footprint, meaning, once a node covers a small enough area. However, we propose to steer cut-off based on the energy. This way, dim nodes (even the ones with a large footprint) will be drawn as patches. The simplest approach would be using a fixed user-defined energy threshold. We instead propose to use a percentage of the *total energy* of the shallowest nodes that completely fit inside the view boundaries. For instance, if the whole dataset fits inside the view boundaries, the shallowest node is the root, and the brightest stars are drawn until a user-defined percentage of the energy of the whole dataset is reached. The *total energy* is computed by summing up the node's stars total energy with the total subtree energy. The advantage of an energy-based cut-off is that brighter nodes are prioritized for visualization and traversal over nodes that would not contribute to the overall energy of the image. Refer to the supplementary materials for additional results on the cut-off.

Spectral accumulation. The star spectra are accumulated in a 3D buffer, where the third axis is used as wavelength range buckets for the energy flux. Stars are fetched from the cache and their spectral histograms are added to a pixel according to their projected sky position. For progressive, interactive rendering, the accumulating into the 3D buffer is done in multiple stages, rather than writing all the spectrum simultaneously. For better visual effects, the accumulation is done in strides (e.g., writing to bins 0, 5, and 10 in the first stage instead of bins 0, 1, and 2). As for patches, We note that the patch spectra are not accumulated in the 3D buffer for two reasons; Firstly, the atomic write operator is more expensive than reading, and writing the same spectrum over a region is not computationally efficient. Secondly, the tree does not contain an unreasonably large amount of nodes that need out-of-core rendering techniques. Therefore, we combine the patches and star spectra on the fly: patches are first projected and rasterized to only store the patch IDs at every pixel they cover in the 3D buffer, if any. In case of patch overlaps in some projection types, the 3D buffer contains a set of patch ID slots rather

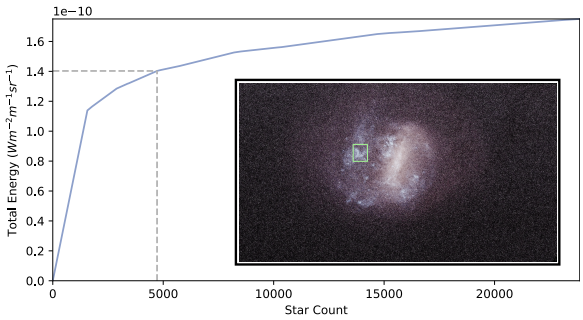


Figure 9: *The Large Magellanic Cloud, one of the closest galaxies to ours, is rendered in our framework. The colors are obtained from the photometry information; the RGB channels contain the G_{RP}, G_G, and G_{BP} magnitude, respectively. The convergence graph of the total flux of the region selected in green as a function of the stars loaded in that region.* 80% *of the energy was obtained after loading* 20% *of the stars.*

than only one layer. Then, the number of pixels each patch covers is computed and stored in a buffer for accurate uniform distribution.

Spectral manipulation. To retrieve a pixel's spectrum, the corresponding histogram from the 3D spectral buffer is added to the spectral patch after dividing it by its area. Then, spectral processing is performed to conduct astronomical analysis and reveal new insights. For instance, applying Eq. 3 to correct for extinction, or detecting emission lines of an element of interest by narrow-band filtering [T*15, WCFJ23]. This step has the same runtime as spectral visualization because the spectra are already accumulated in the spectral image, so the processing time only depends on the number of bins that represent the spectrum.

Spectral visualization and exploration. The final color is computed for each processed spectrum according to the user-defined normalization, tone mapping, image processing, and selected colormap settings. Since mapping from the high-dimensional spectrum to the three-dimensional RGB color is not straightforward, various visualization schemes are employed as discussed in Sec. 3.1. Such as convoluting with CIE sensitivity functions, colormap convolution, 1D colormap, and mapping RP, G, and BP photometry into RGB channels. Lastly, the color is mapped to a low dynamic range suitable for conventional screens.

5 Results and Evaluation

To evaluate the performance of our algorithm and to prove its potential possibilities in the field of astronomy, we implemented

Lactea, an open-source tool able to explore the whole spectral information of GAIA dataset interactively, and allowing the user to choose between multiple visualization modes, both using physical color convolutions as described (CIE, Plank's law) or using arbitrary color maps defined by transfer functions. To the best of our knowledge, this is the first work tackling real-time spectral visualization of large astronomical data. Other systems that visualize GAIA, such as OpenSpace [BAC*20], GaiaSky [SJMS19], or Vaex [BV18] do not incorporate spectral data, so a direct comparison with our approach would not be meaningful.

5.1 Implementation

The data were first downloaded from the GAIA server and processed using a Python script. The continuous XP spectra are converted into sampled histograms using GaiaXPy [Gai] with the same configuration as the released sampled subset discussed in Sec. 3.2. The sampled XP spectra are finally merged with the other attributes from the main GAIA source (source ID, sky position, photometry, temperature, and extinction coefficient) and stored in double precision, conforming our own GAIA sources database we called *StarDB*. The full data contains $219,197,643$ stars with a total energy of $3.68e-06\ Wm^{-2}nm^{-1}$, occupying $629.5GB$ of memory. Subsequently, a C++ code generates the data structure. Each node contains traversal information, the bounding box, and a maximum of N stars. The N stars are then saved in a file, and the node's spectra are saved in a separate file containing N node's subtree spectra for quicker access while rendering patches. N was set to 1000 to generate files of size $5.6MB$, suitable for fast progressive rendering. The total number of nodes is $315,751$, organized in 24 tree levels.

For accessibility, the visualization algorithm is implemented in JavaScript and WebGPU API. Atomic operations are implemented using the atomicCAS (Compare And Swap) [ato24]. For the Infovis part, ChartJS [cha] and D3 [BOH11] are used, and the UI is partially implemented in lilgui [lil]. As commented in Sec. 7, currently WebGPU cannot work with double-precision variables. therefore, SoftFloat [Joh] representation conforming to the IEEE Standard for Floating-Point Arithmetic [Kah96] is implemented in WebGPU Shading Language (WGSL) to work around this issue. Overall, the tree traversal is done in the CPU, while the projection, spectral accumulation, astronomical tasks, and visualization all have their dedicated GPU shaders.

Each pixel on the screen has an accompanying spectrum, which results in a 3D Spectral buffer. The z-axis in the buffer contains 707 4-byte unsigned integers as follows. The XP spectrum is stored in 343 slots, and the extinction-corrected spectrum is stored in the next 343. Additional 7 variables store the total flux of both observed and extinction-corrected spectra, 3 variables store the G, BP, and RP photometry, the temperature, the star count, and the temperature count. All of these variables are in soft float representation as discussed above. The next 3 variables are the star's node ID and the latest written star source ID, which is an unsigned long stored in two slots. The last 10 variables store patch IDs to accommodate overlapping patches in some projection schemes like the spherical projection. The caching employs 3 GPU buffers due to the limitation imposed by WebGPU on a buffer's max capacity. The patch cache is set to $2GB$, enough to store every node's subtree spectrum or to the maximum allowed buffer size. Meanwhile, the star cache has two levels: the first stores the stars in the topmost L level, and the second is a Least Recently Used (LRU) cache for the remaining. Both buffers are set to the maximum allowed buffer size. This design choice guarantees that the topmost nodes will always remain in the GPU while the user moves around any part of the scene.

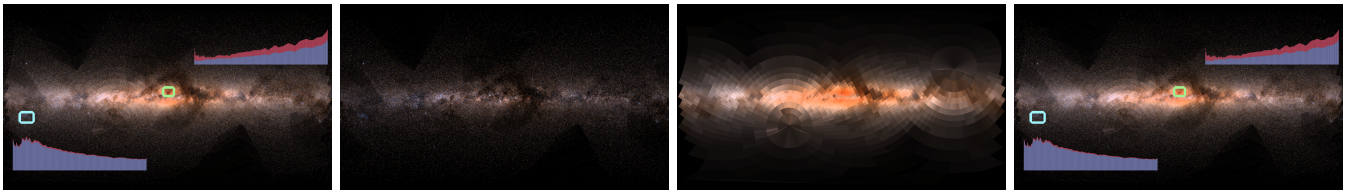
To keep the system interactive and usable in all stages, the compute shader, which accumulates spectra, divides writing a star's spectrum to the spectral buffer into T *tasks* based on the machine's computing capabilities, such as the total stars available in the cache system and WebGPU's maximum allowed workgroup count. Each workgroup processes one star. Each thread in the workgroup processes one bin of the spectrum only, reading the value from the star's spectrum and adding it into the 3D spectral buffer.

The progressive nature of the algorithm allows the system to interactively render the frames while enhancing the quality of the visualization gradually in real-time. However, the time required to achieve granulated visualization depends on the capabilities of the hardware. For instance, on a machine with only $4GB$ of GPU memory, it is possible to allocate $500MB$ for each star cache buffer and a maximum screen resolution of 958×538, occupying $1.5GB$ of memory. In addition, real-time performance can be improved by increasing the number of compute shader tasks.

5.2 Performance evaluation

The tree was built on a cluster running Rocky Linux 9.1 with Intel Skylake CPUs. One node with 4 cores and a budget of 700GB of memory is required to build the tree (either physical memory or file mapped in permanent storage). It took approximately 2 hours to build a tree representing the whole dataset. Our rendering framework is evaluated on a workstation with an Intel Xeon Gold 6230R Processor, 10TB of hard drive space, 252 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of RAM. Evaluation is performed on Ubuntu 20.04 using Google Chrome 131.0.67. The maximum allowed WebGPU buffer size is $4GB$. Therefore, the star cache holds 3,080,000 stars, and the spectral image buffer size is $4GB$ at most. Our system achieves an average of 33 ± 9 fps when moving the camera around, and 50 ± 6 fps otherwise.

We first demonstrate the importance of storing the brighter stars in the top nodes by plotting a region's total energy over time. As the rendering is always progressive, the accuracy of the frame improves progressively in a logarithmic way. Fig. 9 portrays the Large Magellanic Cloud, a dwarf galaxy nearby, as shown by our framework. The tree was traversed until the end without cutting off, so this image is made of $1,451,422$ stars stored in 2604 nodes. The total time it took until convergence is 1.5 minutes. The color is obtained by calculating the G_{RP}, G_G and G_{BP} magnitudes as shown in Sec. 3.2. They are stored in the RGB channels respectively. The values are then divided by the 99th percentile brightness in the image for each channel, rather than the maximum brightness. After normalization, any values greater than 1 are clipped to 1. Finally, log compression [RHD*10] is applied to obtain the final color. Our processing is inspired by the official GAIA DR2 skymap [AMRF20]. Fig. 9 plots the convergence graph of the total energy in the region selected in green as a function of the star count. 80% of the final total energy was obtained after loading 20% of the stars (30 seconds). Refer to the supplementary materials to see the progressive images over time.



(a) No extinction correction (b) Rendering the stars spectra only (c) Rendering the patches spectra only (d) Extinction corrected view

Figure 10: *The whole dataset in galactic coordinates, colored by convoluting with CIE sensitivity function. The full convergence time of the spectral buffer took 1.6 min, while the final renders are fully real-time. This figure shows the star and patch spectral composition:(b) renders the star spectra only, and the nodes at the graph cut are discarded. (c) shows the opposite: only the nodes at the graph cut are rendered as patches in the sky. (a) shows the visualization using the raw spectra: gasses at the center of the galaxies are scattering light, causing a loss of intensity (interstellar extinction), while, finally, (d) is rendered using the extinction-corrected spectra, where some regions are now brighter. The highlighted regions' spectra are blue for the raw spectra and red after correction. The area highlighted in blue is not obscured, so the correction was minimal. In comparison, the green region is amplified.*

The validation of the spectral-preserving aspect is shown in Fig. 10. Fig. 10a is an image of the galactic projection [gai24a] of the whole GAIA catalog. The color is computed in the same way as Fig. 11a. To obtain this image, the tree is cut after 60% of the energy is reached, and the subtree spectrum of the nodes at the graph cut is rendered instead to preserve the energy. The image comprises $1,761,000$ stars loaded from 1761 nodes and the uniformly distributed spectra of 1762 patches. The total time till convergence is 1.6 minutes. Stars are rendered as points, and the subtree spectrum of the node at the graph cut is rendered as patches. The total flux in this image is $3.68e-06\ Wm^{-2}nm^{-1}$, which is the exact total energy of the whole tree, but rendered with loading 0.5% of the data only, thanks to the dual representation in our data structure. Fig. 10b shows the spectral image if the stars are rendered and the nodes at the graph cut are discarded. In contrast, Fig. 10c shows the rendering of the spectra of the patches at the graph cut only. Their composition is the final image in Fig. 10a.

Fig. 11 depicts parts of our galaxy and the Large and Small Magellanic Clouds in spherical projection. To obtain this image, the tree is cut after 80% of the energy is reached, and the subtree spectrum of the nodes at the graph cut is rendered instead to preserve the energy. The image comprises $2,742,000$ stars loaded from 2742 nodes and the uniformly distributed spectra of 2663 patches. The total time until convergence is 1.9 minutes. Fig. 11a shows the same view colored via convolution with CIE sensitivity functions and normalized. The intensity is normalized using the 95% and 1% percentile intensities and then cropping the value to within the $0-1$ range. Then, sigmoid compression is applied. The result is stored in the alpha channel. Fig. 11b is the intensity map using the same steps to obtain the normalized intensities as those stored in the alpha channel in Fig. 11a. The intensity value is then used for color lookup. The spectra of the selected regions are shown in Fig. 11. The region in green is concentrated more on the blue side of the spectrum, while the other region has more intensity in the red part.

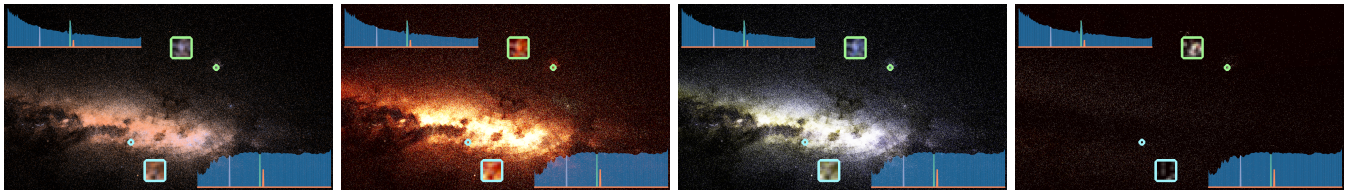
Once the spectral image is built, astronomical tasks can be conducted on top with negligible computation time. For instance, applying mathematical equations to each bin in the spectra is as fast as spectral color computation.

6 Case Studies

We conducted an informal interview with two astrophysicists regarding our proposed method's utility for the analysis of star spectra. They highlighted the value of real-time spectral exploration, noting its potential to support various analysis tasks in their research. The progressive nature of the algorithm allows for interactive navigation over the dataset while keeping the full spectral information available—a process that would be extremely slow with their current workflow and tools—enabling them to explore and discover regions of interest easily. We showed them the *Lactea* tool containing a preliminary version of the transfer function editing, which prompted feedback on how to employ the spectral data in their analysis: The capability of having the whole spectral information in a buffer enables visualizing the information in various ways depending on the use case they need. In particular, they discussed the relevance of spectral visualization for studying interstellar extinction (a phenomenon in which starlight is absorbed or scattered by interstellar dust and gas) observable through characteristic features in the spectra. Additionally, they pointed out the utility of visualizing only spectral lines associated with specific elements, such as hydrogen emission. After studying photometry [SM92, BD07] and spectroscopy [Hea90, Ten19] in more detail, we refined the user interface based on the expert feedback to enable three exemplary use cases, detailed below.

6.1 Interstellar Extinction and Reddening

As explained in Sec. 3.1, extinction occurs when intergalactic gas absorbs and scatters the spectra and reduces light intensity. Blue waves are affected more because they are shorter than red waves. Hence, a reddening effect occurs. We use the Fitzpatrick extinction law [Fit99] in Eq. 3 to compute the extinction corrected spectra. Since each star's extinction coefficient A_0 differs, applying the law directly in the accumulated subtree spectra is not mathematically correct. Therefore, the corrected spectra are stored for each tree node as well. Fig. 10a illustrates the visualization using the raw spectra: the image is overly red, especially at the galaxy center, due to the scattering and absorption of the spectra, so stars appear cooler (therefore redder) than what their surface temperature suggests. Fig. 10d is the same view after correcting the spectra for extinction. Some stars are brighter and a bit bluer than the original image (therefore hotter) due to the correction of lost blue light. The total energy of this image is $4.08e-06\ Wm^{-2}nm^{-1}$, which is higher than the one produced by the raw spectra. The spectra of the highlighted regions in blue and green are shown on top of each figure. The raw spectrum is shown in blue, and the corrected one is in red. The spectrum is barely corrected in the blue region because the starlight was not obscured by anything; hence, its extinction coefficient is close to zero. In comparison, the spectrum of the region highlighted in green is corrected and amplified. As a consequence, the region appears brighter.



(a) CIE sensitivity functions convoluted spectra (b) Intensity map (c) Hubble Palette (d) H_α emission after continuum subtraction

Figure 11: *The Magellanic Clouds and part of the Milky Way in spherical projection. The full convergence time of the spectral buffer took 1.9 min, while the final renders are fully real-time. The colors in (a) is the normalized convolution with CIE sensitivity functions and scaled by the intensity, while (b) is obtained based on the energy. (c) is an example of narrowband filtering and (d) is an example of emission line detection.*

6.2 Narrow-band Filtering

Narrow-band filtering, as introduced in Sec. 3.1, is utilized to obtain a Hubble palette image. Three transmission functions modeled as a Gaussian centered around $SII = 673nm$, $H_\alpha = 656nm$, and $OIII = 500.7nm$ evaluates Eq. 2, obtaining three intensities mapped to the RGB channels, respectively. The final color is processed in the same way as shown in Fig. 11b. The resulting image is shown in Fig. 11c.

6.3 Emission Line Detection

Narrow-band filtering isolates the spectra at a specific region. Still, it does not consider whether the element of interest is emitted at that location or just obtained from the continuous spectra. One solution is to subtract the *continuum spectrum* from the spectrum of interest [Ten19]. Then, narrow-band filters are applied, as explained in the previous case study. The continuum spectrum is the spectrum without emission or absorption lines. There are different ways to compute it, but for simplicity and quick computation, we convolve the spectra with a Gaussian to obtain the continuum, then subtract it from the pixel spectrum, and finally apply narrow-band filtering. Our workflow in this use case is partially driven by Finkbeiner [Fin03], who presented a full-sky H_α map from various star surveys, and Weiler et al. [WCFJ23], who analyzed the continuous XP spectra and presented example applications with hydrogen Balmer lines, among many others. Fig. 11d shows the regions with H_α emission lines. The continuum spectrum is obtained by smoothing the observed spectra with a Gaussian kernel $\sigma = 10$. Then, the continuum is subtracted from the observed spectrum. Lastly, a Gaussian centered around H_α is used as the transmission function in Eq. 2. The region in blue is completely dark compared to Fig. 11b, because H_α in that region is absorbed, not emitted. In comparison, the region highlighted in green has a strong emission line around H_α, and it was matched accordingly.

7 Limitations and Future Work

We have demonstrated the potential of our system for analyzing complex astronomical tasks, showcasing its versatility and foundational capabilities. However, we have used a simplified approach for the computation of the continuum, using the emission curve method of [Fit99]. Nevertheless, our system is adaptable to incorporate more advanced and specialized techniques. Recent methods in the astronomy literature, including those tailored to the GAIA dataset and cutting-edge deep learning approaches [ZYH*24], could seamlessly integrate with our framework. Our work provides a robust foundation for a variety of spectral manipulation and processing tasks. To illustrate our system's strengths, we evaluated our technique on GAIA's XP spectra and photometry, highlighting the spectral-preserving, progressive, and interactive rendering capabilities. Our approach could be extended to handle the higher resolution GAIA RVS spectra discussed earlier, which astronomers analyze to detect metallicity and many other stellar properties. Furthermore, our system could be easily extended to study additional spectral astronomical tasks, such as spectral feature identification could aid in white dwarf identification for characterizing exoplanets [KHR*24]. In addition, the Doppler redshift could be studied in a similar way to the extinction correction demonstrated in this paper.

Even though our technique works theoretically on any precision or resolution, our current implementation has a few technical limitations due to the early state of WebGPU technology, and most of them will be solved with its coming releases. As discussed in Sec. 5.1, a current limitation is due to WebGPU's inability to handle double precision numbers, which can lead to accuracy loss in some extreme zoom-in cases. Another current WebGPU limitation is the limit on the maximum allowed buffer size, which restricts the maximum output resolution. However, implementing a simple tiling strategy would allow arbitrary resolutions, which we leave as future work. We are also planning to add extensions for efficient spectral representation, ideally a mixed model between the original Hermite polynomial XP spectra and the binned RVS spectra, as well as handling and visualizing uncertainties and missing data values.

8 Conclusion

We have presented a novel spectral-preserving, multi-resolution data structure and a progressive, interactive spectral visualization and exploration algorithm. Our technique was validated with the GAIA dataset, dealing with around 630 gigabytes of spectra and other attributes of 200 million stars from the Milky Way. We showed the efficiency of our method and demonstrated three astronomically-motivated use cases of our system. This work is the first to offer spectral visualization and exploration on the GAIA dataset. Ultimately, we hope our open-source implementation might serve as a foundation for various spectrum-related analysis tasks.

Acknowledgments

This work has made use of data from the European Space Agency (ESA) mission *Gaia*, processed by the *Gaia* Data Processing and Analysis Consortium. This research has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project Number 327154368 - SFB 1313. Special thanks go to Jackie Faherty and Brian Abbott for the valuable discussions and advice, to D. Baumgartner, S. Döring, and N. Gärtner for the first prototype, and to Marco Agus for his support. Open access publishing facilitated by King Abdullah University of Science and

Technology, as part of the Wiley - King Abdullah University of Science and Technology (KAUST) agreement.

References

[A*23] ANDRAE R., ET AL.: Gaiadata release 3: Analysis of the gaia bp/rp spectra using the general stellar parameterizer from photometry. *Astronomy & Astrophysics 674* (2023), A27. doi:10.1051/0004-6361/202243462. 2, 4

[ACS*17] AXELSSON E., COSTA J., SILVA C., EMMART C., BOCK A., YNNERMAN A.: Dynamic scene graph: Enabling scaling, positioning, and navigation in the universe. *Computer Graphics Forum 36*, 3 (jun 2017), 459–468. doi:10.1111/cgf.13202. 2

[AFK*14] AMIRKHANOV A., FRÖHLER B., KASTNER J., GRÖLLER E., HEINZL C.: Inspectr: Multi-modal exploration, visualization, and analysis of spectral data. *Computer Graphics Forum 33*, 3 (2014), 91–100. doi:https://doi.org/10.1111/cgf.12365. 4

[AMJV*23] ALGHAMDI R., MÜLLER T., JASPE-VILLANUEVA A., HADWIGER M., SADLO F.: Doppler volume rendering: A dynamic, piecewise linear spectral representation for visualizing astrophysics simulations. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, pp. 39–49. 2

[AMRF20] ALAN MACROBERT R. S., RICK FIENBERG E. W.: Gaia dr2: Esa/gaia/dpac. constellation figures based on those developed for the iau, 2020. URL: https://svs.gsfc.nasa.gov/4851/. 8

[ato24] *CUDA C++ Programming Guide | Release 12.6*. NVIDIA Corporation, 2024. URL: https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf. 8

[BAC*20] BOCK A., AXELSSON E., COSTA J., PAYNE G., ACINAPURA M., TRAKINSKI V., EMMART C., SILVA C., HANSEN C., YNNERMAN A.: Openspace: A system for astrographics. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 633–642. doi:10.1109/TVCG.2019.2934259. 2, 8

[Bal13] BALL R. S.: *Right Ascension and Declination; Celestial Latitude and Longitude*. Cambridge Library Collection - Astronomy. Cambridge University Press, 2013, p. 82–115. 4

[BD07] BUDDING E., DEMIRCAN O.: *Introduction to astronomical photometry*, vol. 6. Cambridge University Press, 2007. 3, 9

[BGR*16] BAINES D., GIORDANO F., RACERO E., SALGADO J., MARTÍ B. L., MERÍN B., SARMIENTO M.-H., GUTIÉRREZ R., LANDALUCE I. O. D., LEÓN I., TEODORO P. D., GONZÁLEZ J., NIETO S., SEGOVIA J. C., POLLOCK A., ROSA M., ARVISET C., LENNON D., O'MULLANE W., MARCHI G. D.: Visualization of Multi-mission Astronomical Data with ESASky. *Publications of the Astronomical Society of the Pacific 129*, 972 (2016), 028001. Publisher: The Astronomical Society of the Pacific. doi:10.1088/1538-3873/129/972/028001. 2

[BLWW20] BAI Y., LIU J., WANG Y., WANG S.: Machine-learning regression of extinction in the second gaia data release. *The Astronomical Journal 159*, 3 (2020), 84. doi:10.3847/1538-3881/ab63d5. 4

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185. 8

[BV18] BREDDELS M. A., VELJANOSKI J.: Vaex: big data exploration in the era of gaia. *Astronomy & Astrophysics 618* (2018), A13. doi:10.1051/0004-6361/201732493. 2, 8

[CBE*20] COSTA J., BOCK A., EMMART C., HANSEN C., YNNERMAN A., SILVA C.: Interactive visualization of atmospheric effects for celestial bodies. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 785–795. 2

[CCM*20] CIURLO A., CAMPBELL R. D., MORRIS M. R., DO T., GHEZ A. M., HEES A., SITARSKI B. N., KOSMO O'NEIL K., CHU D. S., MARTINEZ G. D., NAOZ S., STEPHAN A. P.: A population of dust-enshrouded objects orbiting the galactic black hole. *Nature 577*, 7790 (2020), 337–340. doi:10.1038/s41586-019-1883-y. 1

[cha] CHARTJS TEAM: Chart.js: Simple yet flexible javascript charting library for the modern web. https://www.chartjs.org/. [Online]. 8

[CO17] CARROLL B. W., OSTLIE D. A.: *An Introduction to Modern Astrophysics*, 2 ed. Cambridge University Press, 2017. 3

[CV18] CASAGRANDE L., VANDENBERG D. A.: On the use of gaia magnitudes and new tables of bolometric corrections. *Monthly Notices of the Royal Astronomical Society: Letters 479*, 1 (2018), L102–L107. doi:10.1093/mnrasl/sly104. 4

[CWJ*21] CARRASCO J. M., WEILER M., JORDI C., FABRICIUS C., DE ANGELI F., EVANS D. W., VAN LEEUWEN F., RIELLO M., MONTEGRIFFO P.: Internal calibration of gaia bp/rp low-resolution spectra. *Astronomy& Astrophysics 652* (2021), A86. doi:10.1051/0004-6361/202141249. 4

[DA*23] DE ANGELI F., ET AL.: Gaiadata release 3: Processing and validation of bp/rp low-resolution spectral data. *Astronomy & Astrophysics 674* (2023), A2. doi:10.1051/0004-6361/202243680. 4

[DBRD*18] DANIELSKI C., BABUSIAUX C., RUIZ-DERN L., SARTORETTI P., ARENOU F.: The empirical gaia g-band extinction coefficient. *Astronomy & Astrophysics 614* (2018), A19. doi:10.1051/0004-6361/201732327. 3

[Eur23] EUROPEAN SPACE AGENCY: Gaia's decade of discoveries: unravelling the intricacies of our galaxy. https://www.esa.int/Science_Exploration/Space_Science/Gaia/Gaia_s_decade_of_discoveries_unravelling_the_intricacies_of_our_galaxy, 2023. [Online]. 1

[Fin03] FINKBEINER D. P.: A full-sky hα template for microwave foreground prediction. *The Astrophysical Journal Supplement Series 146*, 2 (jun 2003), 407. doi:10.1086/374411. 10

[Fit99] FITZPATRICK E. L.: Correcting for the effects of interstellar extinction. *Publications of the Astronomical Society of the Pacific 111*, 755 (1999), 63–75. doi:10.1086/316293. 3, 4, 9, 10

[Gai] Gaiaxpy: The gaia bp/rp spectra package. https://gaia-dpci.github.io/GaiaXPy-website/. [Online]. 8

[gai24a] *Transformations of astrometric data and error propagation*. European Space Agency, 2024. URL: https://gea.esac.esa.int/archive/documentation/GDR3/Data_processing/chap_cu3ast/sec_cu3ast_intro/ssec_cu3ast_intro_tansforms.html. 9

[gai24b] *Use of BP and RP spectra in CU8*. European Space Agency, 2024. URL: https://gea.esac.esa.int/archive/documentation/GDR3/Data_analysis/chap_cu8par/sec_cu8par_data/ssec_cu8par_data_xp.html. 4

[GAM*18] GROSSSCHEDL J. E., ALVES J., MEINGAST S., ACKERL C., ASCENSO J., BOUY H., BURKERT A., FORBRICH J., FÜRNKRANZ V., GOODMAN A., ET AL.: 3d shape of orion a from gaia dr2. *Astronomy & Astrophysics 619* (2018), A106. 1

[GM04] GOBBETTI E., MARTON F.: Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Comput. Graph. 28*, 6 (dec 2004), 815–826. doi:10.1016/j.cag.2004.08.010. 2, 5

[GP11] GROSS M., PFISTER H.: *Point-based graphics*. Elsevier, 2011. 2

[Gre04] GREENBERG J. M.: Creating the "pillars": Multiple meanings of a hubble image. *Public Understanding of Science 13*, 1 (2004), 83–95. doi:10.1177/0963662504042693. 3

[Hea90] HEARNSHAW J. B.: *The analysis of starlight: one hundred and fifty years of astronomical spectroscopy*. CUP Archive, 1990. 3, 9

[HH21] HARRE J.-V., HELLER R.: Digital color codes of stars. *Astronomische Nachrichten 342*, 3 (2021), 578–587. doi:https://doi.org/10.1002/asna.202113868. 2, 4

[HK82] HENDEN A. A., KAITCHUCK R. H.: Astronomical photometry. *New York: Van Nostrand Reinhold* (1982). 3, 4

[HM24] HEER J., MORITZ D.: Mosaic: An architecture for scalable & interoperable data views. *IEEE Transactions on Visualization and Computer Graphics 30*, 1 (2024), 436–446. doi:10.1109/TVCG.2023.3327189. 2

[JGC*10] JORDI C., GEBRAN M., CARRASCO J. M., DE BRUIJNE J., VOSS H., FABRICIUS C., KNUDE J., VALLENARI A., KOHLEY R., MORA A.: Gaiabroad band photometry. *Astronomy & Astrophysics 523* (2010), A48. doi:10.1051/0004-6361/201015441. 4

[Joh] JOHN R. HAUSER: Berkeley softfloat. http://www.jhauser.us/arithmetic/SoftFloat.html. [Online]. 8

[K*17] KARTTUNEN H., ET AL.: *Fundamental Astronomy*. Springer-Verlag, 2017. 3

[Kah96] KAHAN W.: Ieee standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE 754*, 94720-1776 (1996), 11. 8

[KHA12] KAEHLER R., HAHN O., ABEL T.: A novel approach to visualizing dark matter simulations. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2078–2087. 2

[KHR*24] KAO M. L., HAWKINS K., ROGERS L. K., BONSOR A., DUNLAP B. H., SANDERS J. L., MONTGOMERY M. H., WINGET D. E.: Hunting for polluted white dwarfs and other treasures with gaia xp spectra and unsupervised machine learning. *The Astrophysical Journal 970*, 2 (2024), 181. doi:10.3847/1538-4357/ad5d6e. 10

[lil] LIL-GUI TEAM: lil-gui. https://lil-gui.georgealways.com/. [Online]. 8

[LYA*21] LAN F., YOUNG M., ANDERSON L., YNNERMAN A., BOCK A., BORKIN M. A., FORBES A. G., KOLLMEIER J. A., WANG B.: Visualization in astrophysics: Developing new methods, discovering our universe, and educating the earth. *Computer Graphics Forum 40*, 3 (2021), 635–663. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14332, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14332, doi:https://doi.org/10.1111/cgf.14332. 2

[M*17] MOITINHO A., ET AL.: Gaia data release 1 - the archive visualisation service. *A&A 605* (2017), A52. doi:10.1051/0004-6361/201731059. 2

[MHL*18] MARSTON A., HARGIS J., LEVAY K., FORSHAY P., MULLALLY S., SHAW R.: Overview of the mikulski archive for space telescopes for the james webb space telescope data archiving. In *Observatory Operations: Strategies, Processes, and Systems VII* (2018), vol. 10704, SPIE, pp. 416–428. 2

[PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016. 4

[RB*23] RECIO-BLANCO A., ET AL.: Gaiadata release 3: Analysis of rvs spectra using the general stellar parametriser from spectroscopy. *Astronomy & Astrophysics 674* (2023), A29. doi:10.1051/0004-6361/202243750. 4

[RGM*12] RODRIGUEZ M. B., GOBBETTI E., MARTON F., PINTUS R., PINTORE G., TINTI A.: Interactive exploration of gigantic point clouds on mobile devices. In *VAST* (2012), Citeseer, pp. 57–64. 2

[RHD*10] REINHARD E., HEIDRICH W., DEBEVEC P., PATTANAIK S., WARD G., MYSZKOWSKI K.: *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann Series in Computer Graphics]. Elsevier Science, 2010. 8

[RKR*97] RALCHENKO Y., KRAMIDA A., READER J., ET AL.: Nist atomic spectra database, 1997. 3

[S*16] SCHÜTZ M., ET AL.: Potree: Rendering large point clouds in web browsers. *Technische Universität Wien, Wiedeń* (2016). 2

[SJMS19] SAGRISTÀ A., JORDAN S., MÜLLER T., SADLO F.: Gaia sky: Navigating the gaia catalog. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 1070–1079. doi:10.1109/TVCG.2018.2864508. 2, 8

[SM92] STERKEN C., MANFROID J.: *Astronomical photometry: a guide*, vol. 175. Springer Science & Business Media, 1992. 9

[SOW20] SCHÜTZ M., OHRHALLINGER S., WIMMER M.: Fast out-of-core octree generation for massive point clouds. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 155–167. 2

[T*15] TRAVEN G., ET AL.: Thegaia-eso survey: Catalogue of hα emission stars. *Astronomy & Astrophysics 581* (2015), A52. doi:10.1051/0004-6361/201525857. 7

[tea16] *Gaia Collaboration* ET AL.: The Gaia mission. *Astron. Astrophys. 595* (2016), A1. 1

[tea23] *Gaia Collaboration* ET AL.: *Gaia* data release 3. *Astron. Astrophys. 674* (2023), A1. 1, 4

[Ten19] TENNYSON J.: *Astronomical Spectroscopy: An Introduction to the Atomic and Molecular Physics of Astronomical Spectroscopy*. World Scientific, 2019. 9, 10

[VE20] VERBRAECK A., EISEMANN E.: Interactive black-hole visualization. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 796–805. 2

[VE21] VERBRAECK A., EISEMANN E.: Interactive black-hole visualization. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 796–805. doi:10.1109/TVCG.2020.3030452. 2

[WCFJ23] WEILER M., CARRASCO J. M., FABRICIUS C., JORDI C.: Analysing spectral lines in gaia low-resolution spectra. *Astronomy & Astrophysics 671* (2023), A52. doi:10.1051/0004-6361/202244764. 4, 7, 10

[ZGR23] ZHANG X., GREEN G. M., RIX H.-W.: Parameters of 220 million stars from Gaia BP/RP spectra. *Monthly Notices of the Royal Astronomical Society 524*, 2 (06 2023), 1855–1884. doi:10.1093/mnras/stad1941. 4

[Zom06] ZOMBECK M. V.: *Handbook of space astronomy and astrophysics*. Cambridge University Press, 2006. 4

[ZYH*24] ZHANG R., YUAN H., HUANG B., WANG T., YANG L., GREEN G. M., ZHANG X.: An empirical extinction curve revealed by gaia xp spectra and lamost. *The Astrophysical Journal 971*, 2 (2024), 127. doi:10.3847/1538-4357/ad613e. 4, 10