Making Astrometric Solver Tractable through In-Situ Visual Analytics

Konstantin Ryabinin[1][0000−0002−8353−7641], Wolfgang Löffler[1][0009−0003−1319−5601], Olga Erokhina[1][0000−0002−2207−7979], Gerasimos Sarras[1][0009−0005−5316−4062], and Michael Biermann[1][0000−0002−5791−9056]

Astronomisches Rechen-Institut, Center for Astronomy of Heidelberg University, Mönchhofstr. 12–14, 69120 Heidelberg, Germany
konstantin.riabinin@uni-heidelberg.de, gerasimos.sarras@uni-heidelberg.de, loeffler@ari.uni-heidelberg.de, olga.erokhina@uni-heidelberg.de, biermann@ari.uni-heidelberg.de

Abstract. In astronomy, precise determination of stellar positions, proper motions, and parallaxes based on space telescope observations is a Big Data problem. It requires a dedicated software solver running on a high-performance computer to analyse billions of input data records and produce an output stellar catalogue. The solution process relies on a sophisticated model to calibrate out the distortions, which are inevitably presented in the raw input data due to the imperfections of the telescope. After the solution is calculated, its quality must be assessed for physical correctness, scientific value, and possible ways of calibration model improvement. The tools for the solution quality assessment are as important as the solver itself and contribute to the solver's tractability by unveiling the path to fine-tuning the solving process. In our previous work, we created a high-performance astrometric solver AJAS suited for the Japan Astrometry Satellite Mission for INfrared Exploration (JASMINE). In the present work, we foster AJAS tractability by integrating it with the ontology-driven visual analytics platform SciVi leveraging the principles of multi-purpose ontology-driven API for in-situ data processing. This integration provides users with high-level management tools for AJAS computation jobs and high-level visual data mining tools for AJAS solutions. All these tools can be configured via a graphical user Web interface, extended in Jupyter Notebooks, and executed on the same computing resource as AJAS, which minimises the data transfer. In this paper, we elaborate on the technical details of the above-mentioned tools and demonstrate their capabilities on the real examples of the AJAS solution quality assessment.

Keywords: Astrometry · Data Fitting · Visual Analytics · Ontology Engineering · Big Data.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

1 Introduction

Astrometry is a branch of astronomy aiming to precisely determine stellar parameters like position, proper motion, and parallax based on raw telescope observations. Accurate values of these parameters are crucial for for fundamental astronomical and astrophysical studies of the structure and history of the Milky Way Galaxy. [7].

An astrometric solver is a complex software pipeline tackling a Big Data problem with uncertainties. It reveals the stellar parameters based on a massive amount of observational data, which contain uncertainties in the form of various distortions and noise caused by the imperfections of the telescope optics, detector electronics, etc. To reach its scientific aim, an astrometric solver must be performant, numerically accurate, and provide tools for assessing and alleviating uncertainties propagated from the input data to the output.

In our research work, we contribute to the development of the astrometric solver for the Japan Astrometry Satellite Mission for INfrared Exploration (JASMINE) [10]. For JASMINE, a thorough exploration of the crowded area of the Milky Way centre is planned, which involves about 115 thousand stars with 9.2 billion observations. For this purpose we have developed at the Institute for Computational Astronomy (Astronomisches Rechen-Institut, ARI) of Heidelberg University, Germany, the ARI JASMINE Astrometric Solver (AJAS) suited for massively parallel high-performance computers [14,15].

For AJAS, we managed to solve the Big Data processing issue by proposing a dedicated software architecture and utilising state-of-the-art approaches to build high-performance applications. This allows AJAS to handle the expected full-scale JASMINE mission's data within 8.5 hours on a cluster with 5000 CPU cores. In this paper, we propose an approach of making the AJAS tractable by the in-situ visual analytics platform SciVi [4] allowing for monitoring and assessing the solution outcome.

The key contributions of the paper are the following:

1. Bridging astrometry with in-situ visual analytics.
2. Developing the in-situ toolset for assessing the astrometric solution quality for AJAS.
3. Extending the smart interoperability of the SciVi visual analytics platform to better handle the in-situ processing cases.
4. Developing a high-performance data access library for AJAS solutions.

2 Methodology

In general, an astrometric observation o is the centroid with particular coordinates $(\kappa; \mu)$ of a stellar image taken by a telescope at a particular moment of time. The set of observations can be expressed as a function $\boldsymbol{o} = f(\boldsymbol{p})$, where $\boldsymbol{p}$ is a vector of model parameters describing, on the one hand, the best knowledge of the locations and motions of the stars on the celestial sphere, and, on the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

other hand, the best knowledge of the spatial orientation and imaging properties of the telescope used for making these observations. The function f has a complicated nature incorporating a lot of effects including, for example, relativistic effects such as aberration and light bending as well as image distortions of the optical and electronic imaging system. Therefore, f is not fully known and cannot be expressed analytically nor inverted to straightforwardly get $\boldsymbol{p}$. Instead, an optimisation task arises to find the best fit of $\boldsymbol{p}$ by minimising the residuals $\boldsymbol{o} - \boldsymbol{c}$, where $\boldsymbol{c} = g(\boldsymbol{p})$ is a vector of predicted observations and g is a model of f. To adequately model the unknown f function by the g function, a calibration model is introduced that is supposed to describe the differences between the imaging properties of the ideally specified nominal telescope and the actually implemented telescope. The calibration model spawns its own nuisance parameters, which are included in the optimisation process as an integral part of $\boldsymbol{p}$.

The data fitting procedure is based on the least squares approach involving linearisation of g and subsequent solving of the linear equations system. This procedure has two major problems to tackle. The first problem is a Big Data issue concerning the size of the system. It depends on the particularities of the astrometric mission, but even for relatively small missions, the number of equations goes into the billions. The second problem is the quality of the calibration model. To approximate f by g within the target accuracy, the calibration model needs to be properly fine-tuned. These two problems impose challenging requirements the software astrometric solver should meet: it should be very high-performant, flexible, and tractable. It should also contain tools to assess the accuracy and correctness of the solution, and to appraise the ability of the calibration model to absorb systematic distortions.

3 Related Work

The JASMINE astrometric problem is formulated adopting the experience from the ESA Hipparcos [5] and ESA Gaia [7] space astrometry missions and solved by AJAS utilising the direct approach [14] similarly to the Gaia One Day Astrometric Solution (ODAS) [11]. This means that the system of equations for the least squares data fitting is solved in one go, without iterations, by inverting its reduced normal matrix $\boldsymbol{\mathcal{M}}$. Since $\boldsymbol{\mathcal{M}}$ by its nature is rank-deficient [14], AJAS leverages singular value decomposition to calculate its pseudo-inverse $\boldsymbol{\mathcal{M}}^+$. Computations rely on the state-of-the-art libraries ScaLAPACK [1] and EigenExa [16] for number crunching and MPI for inter-process communication. Along with that, hand-crafted optimisations for matrix operations are implemented by considering the peculiarities of the matrix structure and fine-tuning for data locality, efficient multithreading, CPU cache usage, and vectorisation [14].

The JASMINE problem is driven by data. This means that data access and storage are the main bottlenecks of the solver. While running, AJAS scans hundreds of gigabytes and produces terabytes of data, moreover the general data access pattern is highly random and very cache-unfriendly by its nature. A thor-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

ough inspection of the AJAS solution requires access to all the data generated during the calculations. For the traditional posthoc analysis made in software like TOPCAT [17], which is very popular among astronomers, it would be necessary to download the corresponding set of files from the cluster to gain local access. Even the bare minimum subset for the meaningful analysis (for example, the array of solution residuals) is more than 100 gigabytes. This, in turn, would mean spending drastically more time for network communication, than the solving process takes, and potentially abusing the connection channel. Besides that, handling the required data volume locally imposes severe requirements on the performance of a local machine. The natural way to tackle this problem is going for the so-called *in-situ* techniques, which "attempt to avoid the overhead of fully loading and indexing the data in a database management system and improve performance by progressively building an index during data exploration" [12].

In-situ visualisation and analytics is a broad umbrella term that encloses the entire paradigm of processing data as it is generated [2,3]. H. Childs et al. propose an elaborated taxonomy of in-situ systems and define classification criteria to derive a dedicated type for an arbitrary in-situ system [2]. The main idea of in-situ processing is twofold: the data can be processed before their generation is finished, and data transferring overhead is minimised.

Recently, one of the popular and flexible ways of organising in-situ data processing is Jupyter running on the side of a high-performance computer and exposing a Python interpreter to the user via a Web browser [9,18]. Analysis within Jupyter Notebooks gives all the freedom of Python scripting but requires corresponding programming skills from the scientist. In contrast, scientific visualisation systems like ParaView and VisIt provide in-situ processing capabilities with a high-level graphical user interface, smoothing the learning curve but constraining the scientists by a predefined set of analytical tools [3].

To balance between these two approaches, we leverage the in-situ visual analytics for AJAS with the ontology-driven platform SciVi [4,13]. SciVi can run in userspace on high-performance computers, exposes a Web interface, and provides an intuitive visual programming language based on data flow diagrams for defining visual analytics pipelines using a set of predefined operators. The defined pipelines can then be either executed directly within SciVi or automatically transformed into the Jupyter Notebooks. The latter allows for the extension of SciVi operators in Python should they not be enough for advanced analytics beyond the main expected analytical scenarios. Along with that, it is very easy to extend SciVi with new operators written in Python, C++, and JavaScript, making them immediately available in the SciVi Web GUI.

SciVi also can be seamlessly integrated with AJAS supplying settings parameters directly to AJAS modules and allowing the monitoring of intermediate AJAS state on the fly.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

4 AJAS Job Management

Typically, AJAS runs on a CPU cluster as a job that is first submitted to the specific execution queue. The job defines the requested resources (including amount of RAM, number of cluster nodes and CPU cores, time limit, etc.), the runtime environment, and the configuration of AJAS (including paths to input and output data, multithreading parameters, etc.). Traditionally, a job is described and submitted manually, which requires an understanding of the cluster architecture and knowledge of the specific notation supported by the particular queue manager. The monitoring of the execution is then also manual by requesting the job status, while no information about the actual AJAS progress can be retrieved. In this sense, the job submission preparations might be tedious and the tractability of AJAS is limited.

The SciVi system overcomes this hurdle. For AJAS and the cluster, on which AJAS should run, ontological profiles are created and saved in the SciVi knowledge base. The AJAS ontological profile (see Fig. 1) describes the AJAS settings and output data types. The cluster ontological profile (see Fig. 2) describes resource limits and available queue types, as well as provides a job template for the queue manager and the job status retrieving commands. By parsing these profiles, SciVi automatically builds an intuitive user interface for creating the AJAS jobs and monitoring their state. Only high-level settings are exposed to the interface, for example, paths to input and output data and the type of queue to submit the job to. The low-level settings like multithreading options ("Number of Building Threads" and "Number of Summation Threads") are calculated automatically to gain maximal performance. The process grid ("Number of Processes" and "Number of CPU Cores per Process") and time limit settings are customizable but SciVi automatically calculates default values for them based on the chosen input data and queue type. The formulas to calculate the defaults are a part of the AJAS ontological profile (contained in the "AJAS Job" implementation). The service outputs "Start Date" and "Progress" allow monitoring of the particular AJAS job status in realtime.

5 AJAS Solution Analysis

Another part of the demanded AJAS tractability is tuning the solving process and the calibration model based on the solution quality assessment.

The solution analysis consists of two main steps: automatic generation of a standard report and custom data mining. The standard report contains a set of visual and numerical metrics, which, based on our experience, are needed to estimate the solution quality. Custom data mining relies on the interactive SciVi capabilities. The data mining is applicable only if a standard report indicates some solution problems, and it aims to unveil the causes of these problems.

The list of metrics for the standard report and the palette of tools for the custom data mining are still incomplete and are a matter of extension when the real JASMINE mission data will be available. So, here we describe only

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

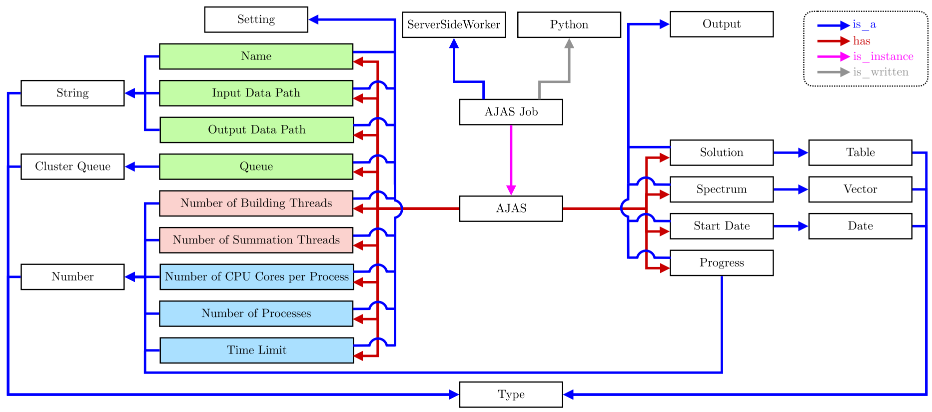

Fig. 1: A fragment of AJAS ontological profile. For the nodes, green highlights the settings given by the user, blue highlights customizable settings which have computable default values, red highlights settings which are determined automatically.

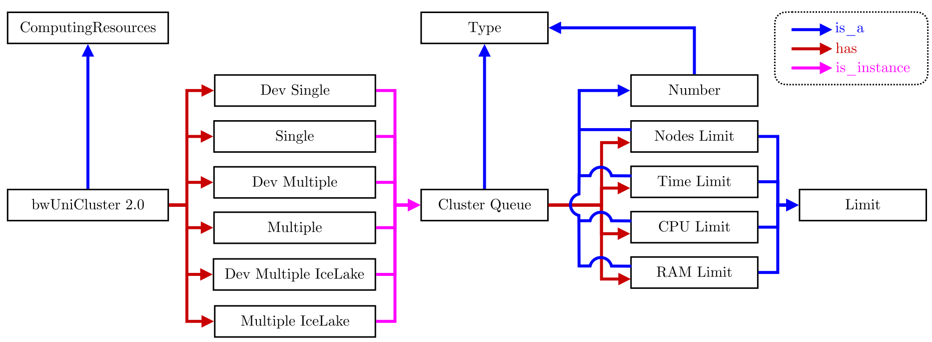

Fig. 2: A fragment of bwUniCluster 2.0 (the cluster available for academic use for the universities of Baden-Württemberg, Germany) ontological profile.

three items to demonstrate the idea of our visual analytics pipeline: plotting the spectrum of $\boldsymbol{\mathcal{M}}$, fitting the Gaussian function to the distribution of astrometric residuals, and plotting the map of uncertainties in the stellar parameters. The presented analytical examples are demonstrated on the test cases, which contain 10 thousand stars and 800 million observations and are solved for two astrometric parameters (stellar position).

5.1 Plotting the Spectrum of $\boldsymbol{\mathcal{M}}$

The mathematical foundation of AJAS has been elaborated in [14]. The main idea is to solve the linearised system

$$\boldsymbol{\mathcal{D}x} = \boldsymbol{o} - \boldsymbol{c}, \tag{1}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

where $\boldsymbol{\mathcal{D}} = (\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{S}})$, $\boldsymbol{\mathcal{C}}$ is the matrix of the Jacobian derivatives of the calibration model, $\boldsymbol{\mathcal{S}}$ is is the matrix of the Jacobian derivatives of the stellar parameters, $\boldsymbol{o}$ is the vector of observations, $\boldsymbol{c}$ is the vector of predicted observations. These predictions are made for the given observation model represented by the derivatives in $\boldsymbol{\mathcal{D}}$ and the initial guess of the calibration and stellar parameters $\boldsymbol{p}$. Then, $\boldsymbol{x}$ is a vector of updates for $\boldsymbol{p}$.

The system (1) is overdetermined and the normal matrix $\boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{D}}^{\mathsf{T}}\boldsymbol{\mathcal{D}}$ is rank-deficient, so we use the least squares fitting to resolve $\boldsymbol{x}$. For this, the pseudo-inverse matrix $\boldsymbol{\mathcal{N}}^{+}$ must be found, which is computationally not possible because of the huge size of $\boldsymbol{\mathcal{N}}$. Instead, a reduced normal matrix $\boldsymbol{\mathcal{M}}$ is calculated by forward-eliminating parts of $\boldsymbol{\mathcal{C}}$ [14]. The $\boldsymbol{\mathcal{M}}$ matrix is then almost two orders of magnitude smaller than $\boldsymbol{\mathcal{N}}$ and can be inverted in reasonable time by the singular value decomposition:

$$\boldsymbol{\mathcal{M}}^{+} = \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{E}}^{-1}\boldsymbol{\mathcal{Z}}^{\mathsf{T}}, \tag{2}$$

where $\boldsymbol{\mathcal{Z}}$ is a matrix of eigenvectors and $\boldsymbol{\mathcal{E}}$ is a diagonal matrix of singular values (non-zero eigenvalues) of $\boldsymbol{\mathcal{M}}$.

The eigenvalue spectrum of $\boldsymbol{\mathcal{M}}$ gives information about the degeneracy of the system. Because of its inherent rank deficiency, the eigenvalue spectrum of $\boldsymbol{\mathcal{M}}$ will always contain zero eigenvalues. These can, however, be eliminated by appropriately constraining the system. If not eliminated by constraints, these algebraically but not necessarily numerically zero eigenvalues pose a problem. In addition, more eigenvalues may become numerically small when there are not enough observations to determine the corresponding model parameter, contributing to the same problem. Once inverted in (2), these near zero eigenvalues will become large and numerically destroy the solution. To prevent this, they have to be zeroed out in $\boldsymbol{\mathcal{E}}^{-1}$.

Logarithmic spectrum plots (Fig. 3) help to inspect the consistency of input data and the numerical stability of the solution. For example, the top plot in Fig. 3 shows that the spectrum has several very small eigenvalues spanning the spectrum's dynamic range from 10^{-10} to 10^{8}, which is beyond the range of 64-bit floating point data type. This indicates the numerical instability and degeneracy of the system. The degeneracy comes from the fact that the system has the freedom to either calibrate the observations, bringing them to the predicted stellar positions, or to update the stellar positions, bringing them to the observations. To restrict this freedom, the set of stars is used, whose parameters are known with high precision. These stars come from the Gaia DR3 catalogue [6]. For them, extra summands are put to the $\boldsymbol{\mathcal{S}}$ block of the system's design matrix $\boldsymbol{\mathcal{D}}$ reinforcing the weight of corresponding Jacobian derivatives. The improvement of the spectrum in this case is shown in the bottom plot in Fig. 3.

In some cases, zero and also negative eigenvalues may pop up. To stick with the logarithmic scale, absolute values of eigenvalues are taken, and those, which were negative, are then marked red in the plot and a corresponding legend appears explaining the meaning of colours.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

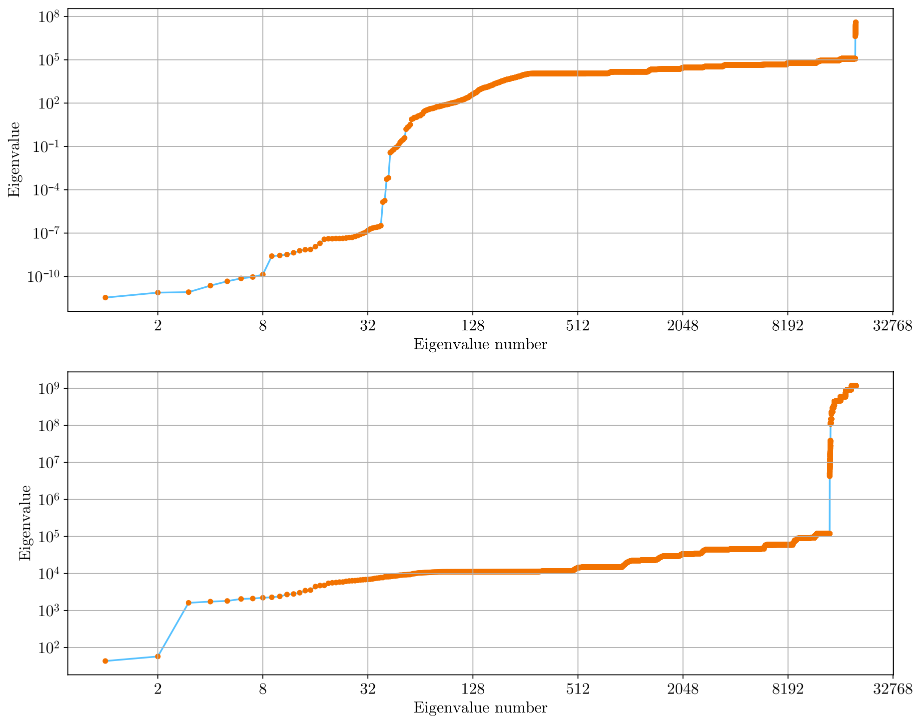

Fig. 3: Plot of the $\boldsymbol{\mathcal{M}}$ spectrum drawn by SciVi using matplotlib [8]. Top: the $\boldsymbol{\mathcal{S}}$ block of the system's design matrix $\boldsymbol{\mathcal{D}}$ is *not* reinforced by the stars from the Gaia DR3 catalogue, bottom: the $\boldsymbol{\mathcal{S}}$ block *is* reinforced.

5.2 Fitting a Gaussian to the Residuals

If all systematic uncertainties have been accounted for by the calibration model, the residuals $\boldsymbol{r} = \boldsymbol{o} - \boldsymbol{c} - \boldsymbol{\mathcal{D}}\boldsymbol{x}$ should have a perfect Gaussian distribution reflecting the remaining, purely random observational noise. If the distribution of $\boldsymbol{r}$ differs from Gaussian, it means, that the utilised calibration model was unable to absorb all the systematic errors of the observations, for example, optical distortions introduced by the telescope, geometrical imperfections of the focal plane, etc. In this case, $\boldsymbol{x}$ is not the desired astrometric solution and the calibration model has to be improved. Finding the way to that improvement is a challenge. The first step in that way is the identification of the observations, which deviate the distribution from the Gaussian. The common characteristics of these observations will then give a hint, which effects are missing in the calibration model.

To allow this type of analysis, the histogram of residuals is built and the Gaussian curve is fitted to it. There are also tools, which provide the possibility to perform this operation on an arbitrary subset of observations. Fig. 4 demonstrates the corresponding visualisation results for two cases. Both plots show

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

residuals in one direction (η) of the field-of-view reference system (FoVRS). The top plot corresponds to the case when the calibration model is unable to absorb all systematic distortions. Two problems are immediately seen: the goodness of fit for the Gaussian curve is not high enough, just 0.97, and, more importantly, the Gaussian distribution is not centred at zero ($\mu = 10^{-9}$). The bottom plot shows the case when all the systematic distortions are removed. The goodness of fit for the Gaussian curve is almost exactly 1.0 and it is centred at zero ($\mu = 10^{-14}$), which reflects the pure random observational noise.

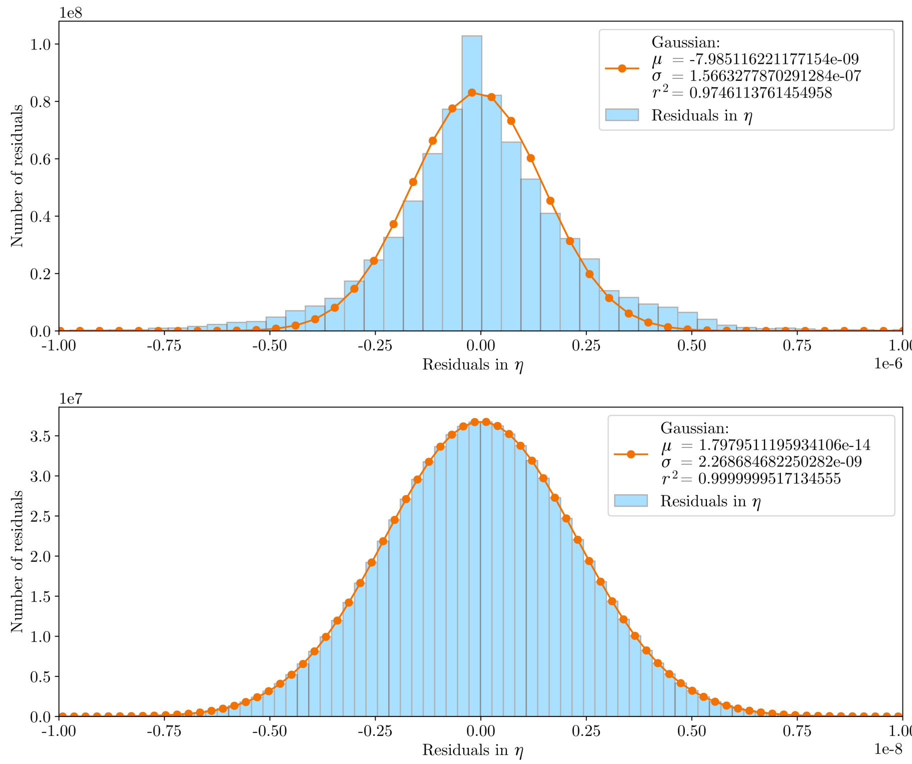

Fig. 4: Fitting the Gaussian curve to the histogram of residuals drawn by SciVi using matplotlib. Top: systematic remains in the residuals, bottom: systematic is fully removed. Note, that the axes of both plots have different scales.

5.3 Plotting the Stellar Uncertainties

The pseudo-inverse matrix $\boldsymbol{\mathcal{M}}^{+}$ is a covariance matrix for the $\boldsymbol{\mathcal{S}}$ block of (1), which means, its main diagonal contains standard errors of the stellar parameters. The square roots of these elements represent corresponding uncertainties,

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

which, in turn, can be used for the solution quality assessment. To visually inspect them, we plot them as a colour-coded stellar map (see Fig. 5). The dots correspond to the stars drawn in the International Celestial Reference System (ICRS), and the colours represent uncertainties of a chosen stellar parameter (for example, one of the position coordinates). This representation way highlights the sky regions, which are covered worse than others by observations. Based on this information, the observation strategy of the satellite can be amended to improve coverage of these problematic regions.

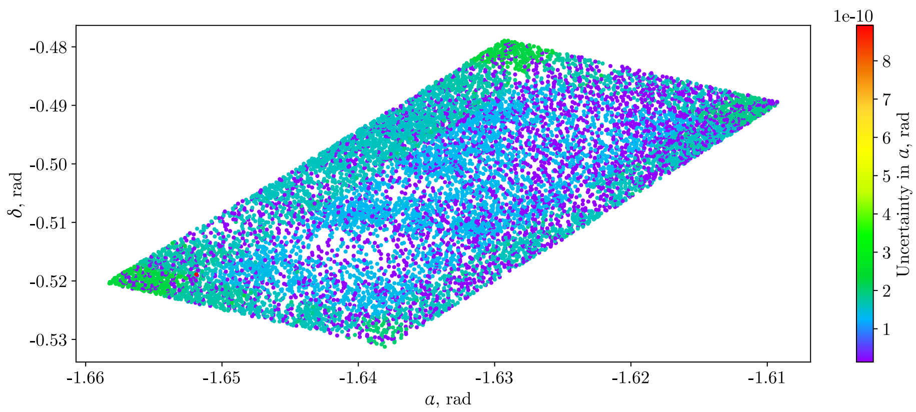

Fig. 5: Map of stellar uncertainties of the first position coordinate in ICRS drawn in SciVi using matplotlib.

In this example, four categories of stars can be visually distinguished according to the uncertainty of the α coordinate (one angular position coordinate in ICRS):

1. Stars with very low uncertainty are highlighted with a violet colour. As mentioned in Section 5.1, these stars have a very good initial guess of their position because they are taken from the Gaia DR3 catalogue and used to resolve the degeneracy between the calibration model and stellar positions. As can be seen in Fig. 5, they are distributed pretty evenly over the JASMINE target region, providing good coverage of the observed field.
2. Stars with acceptably low uncertainty are highlighted with a blue colour. These are the majority of stars inside the JASMINE target region, which have been observed a sufficient number of times to obtain a high-quality astrometric solution.
3. Stars with higher uncertainty are highlighted with a greenish and green colours. These stars are on the border (especially in the corners) of the JASMINE target region, so they are observed fewer times, which slightly worsens the quality of the astrometric solution for them.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

4. A single star located approximately at $(-1.652; -0.519)$ with very high uncertainty is highlighted with a red colour. This star has only 59 observations, while the others have, on average, 77 thousand observations (almost four orders of magnitude more). Therefore, the solution of this star should be used with caution. However, since in this category there is only one star out of 10 thousand observed in this test case, the overall quality of the solution is considered high.

As an improvement for this visualization type, we plan to implement the ability to query and plot corresponding observations for chosen stars on demand. Plotting all observations at once is pointless, but showing individual observations for specific stars at the appropriate zoom level might be helpful for inspecting the astrometric solution.

In the future, we are also interested in finding a presentation form for the off-diagonal elements of $\boldsymbol{\mathcal{M}}^+$, which express the covariances of different stellar and higher-order calibration parameters.

5.4 Organizing the Custom Visual Analytics in SciVi

The standard report gives an overview of the whole solution, but if problems are identified, more fine-grained manual analysis is needed. SciVi allows the user to customize the analytical pipeline defining particular data flow diagrams (DFDs), which declare querying, transforming, and visualizing appropriate subsets of data.

Let us assume for example that the fitting of the Gaussian curve to the whole set of residuals leads to unsatisfactory results. It means that the input data contain systematic distortion that was not absorbed by the calibration model. To identify which observations introduce this distortion, different subsets of the solution should be investigated individually. Fig. 6 demonstrates the DFD describing a custom visual analytics pipeline. The AJAS solution is filtered to extract its subset according to the given criteria specified in the settings of the “Filter” operator. These settings are not shown in the diagram as they are displayed separately in the SciVi user interface when the user clicks on this operator. Then, a histogram is created and a Gaussian curve is fitted to it. Then, both the histogram and the curve are plotted, joined together and displayed to the user.

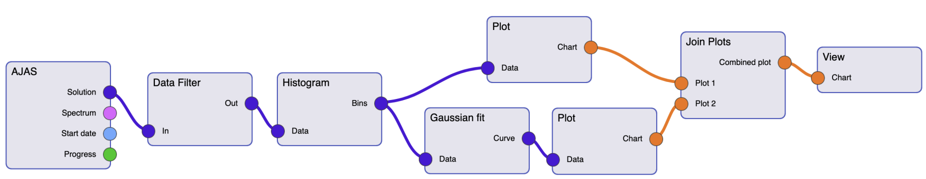

Fig. 6: DFD of a custom visual analytics pipeline in SciVi.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

Each operator has its own ontological profile (like, for example, the one for AJAS is shown in Fig. 1). This profile specifies the inputs, outputs, and settings of the operator along with its implementation and the computing resource it should be executed on. Based on this information, SciVi maintains the graphical user interface of the operator, its appearance in the DFD, and its interoperability with other operators within the pipeline defined by DFD.

In the DFD from Fig. 6, all the operators except for the "View" are specified to run on the server (supercomputer) side to have direct access to the data. The "View" operator combines two actions: rendering of the plot, which is also performed on the server side, and displaying the rendering result to the user, which happens on the client side (in the user's Web browser). The ontological profile of the compound "View" operator is shown in Fig. 7. Here, suboperators "Render" and "Display" are linked to "View" as its parts, and the output of "Render" is declared to be used as an input of "Display", while "Render" is linked to the server side and "Display" is linked to the client side. The data transfer between them is managed by SciVi automatically.

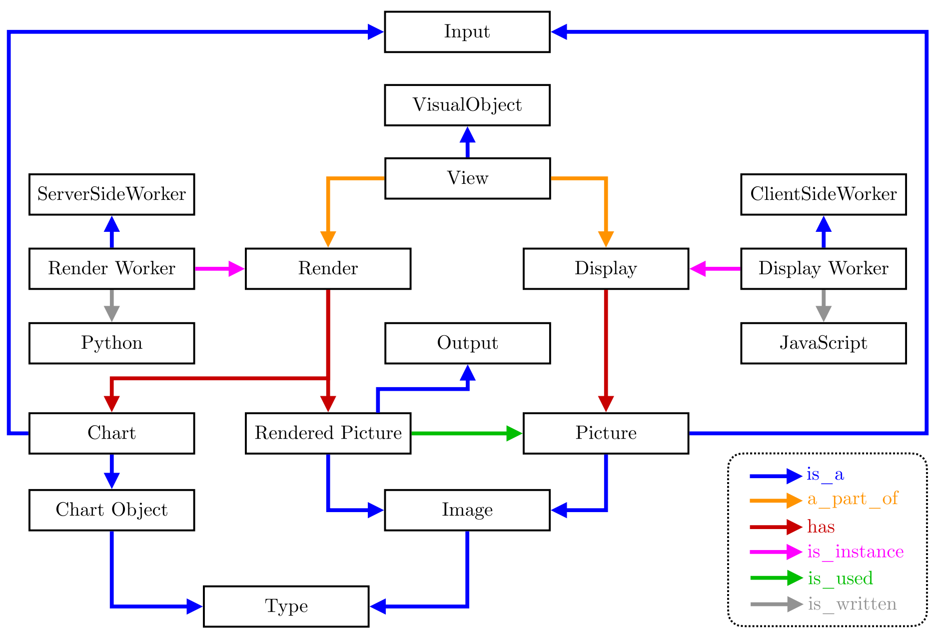

Fig. 7: Ontological profile of the "View" operator.

The concept of compound operators is new to SciVi and is first introduced in this work. It is a further improvement of the smart interoperability introduced in [13]. Smart interoperability allows different operators within the same DFD to run on different computing resources and freely exchange the data with minimal transmission overhead. However, each regular operator is tied to its

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

computing resource. The compound operators generalise the smart interoperability approach to the cases, where a single operator is distributed over several different computing resources. This makes SciVi DFDs more versatile, efficient, and concise.

All the AJAS-related server-side SciVi operators rely on the Rapid ACCess Operations On Numerical Solutions (RACCOONS) library. We developed this library in C++ and created a binding to Python. Its core implements a multithreaded data querying engine and its Python interface provides `pandas`-like access to the AJAS solution. The querying engine supports lazy caching and on-demand indexing of data, which optimises the analytics process when the user issues requests for data of a similar nature.

Since SciVi operators are internally implemented in Python, it was for us straightforward to implement the automatic dumping of any particular SciVi visual analytics pipeline to a Jupyter Notebook. This feature allows the user to customise the pipeline even further and build upon it more complicated processing machinery using Python.

6 Conclusion

We integrate the ontology-driven visual analytics platform SciVi with AJAS using the principles of a multipurpose ontology-driven API and run them on the same computing resource to avoid unnecessary data transfer. Within the SciVi environment, we developed a set of tools providing both automatic and human-in-the-loop operation controls, which enable AJAS tractability via in-situ visualisation and analytics. SciVi exposes these tools via a Web interface and automatically generates for them an intuitive graphical user interface that allows the users to build data processing pipelines using a visual programming language based on data flow diagrams. These tools facilitate starting AJAS on a cluster, monitoring its progress, and extensive analysis of its output. Advanced visual analytics helps to assess the quality of astrometric solutions produced by AJAS, identify the issues and find out the ways to corresponding improvements via the fine-tuning of the AJAS solving process.

Data processing pipelines created in SciVi can be automatically converted into Jupyter Notebooks and then further customized in Python.

We demonstrate the SciVi capabilities on three examples of analysing $\boldsymbol{\mathcal{M}}^{+}$ spectrum, astrometric system's residuals, and astrometric parameters' uncertainties. This list, however, is still incomplete to fully assess the quality of the astrometric solution, so, a part of future work is to extend this toolset with other instruments and metrics. Another direction of improvement is the optimisation of the RACCOONS library that provides efficient access to the AJAS data.

7 Acknowledgments

This work was financially supported by the German Aerospace Agency (Deutsches Zentrum für Luft- und Raumfahrt e.V., DLR) through grant 50OD2201. The authors acknowledge support by the state of Baden-Württemberg through bwHPC. The authors also thank the National Astronomical Observatory of Japan (NAOJ) for fruitful discussions about the mission specifics.

References

1. Blackford, L.S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., Whaley, R.C.: ScaLAPACK Users' Guide. Society for Industrial and Applied Mathematics, Philadelphia, PA (1997)
2. Childs, H., Ahern, S.D., Ahrens, J., Bauer, A.C., Bennett, J., Bethel, E.W., Bremer, P.T., Brugger, E., Cottam, J., Dorier, M., et al.: A terminology for in situ visualization and analysis systems. The International Journal of High Performance Computing Applications **34**(6), 676–691 (2020). https://doi.org/10.1177/1094342020935991
3. Childs, H., Bennett, J.C., Garth, C.: In Situ Visualization for Computational Science. Springer Cham (2022)
4. Chuprina, S., Ryabinin, K., Koznov, D., Matkin, K.: Ontology-Driven Visual Analytics Software Development. Programming and Computer Software **48**, 208–214 (2022). https://doi.org/10.1134/S0361768822030033
5. ESA: The Hipparcos and Tycho Catalogues. ESA SP-1200 (1997), https://www.cosmos.esa.int/web/hipparcos/catalogues
6. Gaia Collaboration, Drimmel, R., Romero-Gómez, M., Chemin, L., Ramos, P., Poggio, E., Ripepi, V., Andrae, R., Blomme, R., Cantat-Gaudin, T., et al.: Gaia data release 3 - mapping the asymmetric disc of the milky way. Astronomy & Astrophysics **674**, A37 (2023). https://doi.org/10.1051/0004-6361/202243797, https://doi.org/10.1051/0004-6361/202243797
7. Gaia Collaboration, Prusti, T., de Bruijne, J.H.J., Brown, A.G.A., Vallenari, A., Babusiaux, C., Bailer-Jones, C.A.L., Bastian, U., Biermann, M., Evans, D.W., et al.: The Gaia mission. Astronomy & Astrophysics **595**, A1 (Nov 2016). https://doi.org/10.1051/0004-6361/201629272
8. Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in Science & Engineering **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55
9. Ibrahim, S., Stitt, T., Larsen, M., Harrison, C.: Interactive in situ visualization and analysis using ascent and jupyter. In: Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization. pp. 44–48. ISAV '19, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3364228.3364232, https://doi.org/10.1145/3364228.3364232
10. Kawata, D., Kawahara, H., Gouda, N., Secrest, N.J., Kano, R., Kataza, H., Isobe, N., Ohsawa, R., Usui, F., Yamada, Y., et al.: JASMINE: Near-infrared astrometry and time-series photometry science. Publications of the Astronomical Society of Japan **76**(3), 386–425 (2024). https://doi.org/10.1093/pasj/psae020, https://doi.org/10.1093/pasj/psae020
11. Löffler, W., Bastian, U., Biermann, M., Jordan, S., Brüsemeister, T., Stampa, U., Bernstein, H.H.: The one-day astrometric solution for the gaia mission. Astronomy & Astrophysics (in preparation)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11

12. Papastefanatos, G., Alexiou, G., Bikakis, N., Maroulis, S., Stamatopoulos, V.: VisualFacts: A Platform for In-Situ Visual Exploration and Real-Time Entity Resolution. In: EDBT/ICDT Workshops (2022), https://api.semanticscholar.org/CorpusID:248893496
13. Ryabinin, K., Chuprina, S., Labutin, I.: Tackling IoT Interoperability Problems with Ontology-Driven Smart Approach. Lecture Notes in Networks and Systems **342**, 77–91 (2021). https://doi.org/10.1007/978-3-030-89477-1_9
14. Ryabinin, K., Sarras, G., Löffler, W., Olga, E., Biermann, M.: AJAS: A High Performance Direct Solver for Advancing High Precision Astrometry. Journal of Computational Science (Accepted for publication in 2025)
15. Ryabinin, K., Sarras, G., Löffler, W., Erokhina, O., Biermann, M.: Satellite telescope self-calibration through precise stellar data mining. Frontiers in Artificial Intelligence and Applications **398**, 248–254 (2024). https://doi.org/10.3233/FAIA241425
16. Sakurai, T., Futamura, Y., Imakura, A., Imamura, T.: Scalable Eigen-Analysis Engine for Large-Scale Eigenvalue Problems, pp. 37–57. Springer Singapore, Singapore (2019). https://doi.org/10.1007/978-981-13-1924-2_3
17. Taylor, M.: TOPCAT: Working with Data and Working with Users (2017), https://arxiv.org/abs/1711.01885
18. Tsai, S.R., Schive, H.Y., Turk, M.: Libyt: A Tool for Parallel In Situ Analysis with yt, Python, and Jupyter. In: Proceedings of the Platform for Advanced Scientific Computing Conference. PASC '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3659914.3659939, https://doi.org/10.1145/3659914.3659939

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97573-8_11