SoftwareX 31 (2025) 102203

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Juniper: A 3D plotting framework for effective multi-dimensional scatterplots

Philippos Papaphilippou

University of Southampton, United Kingdom

ARTICLE INFO

Keywords:
3D scatterplots
Multi-dimensional
Design space
Visual cues
Depth perception

ABSTRACT

Computationally and data intensive workloads including design space explorations or large studies often lead to multi-dimensional results, which are often not trivial to digest with conventional plotting software. 3D scatterplots can be a powerful technique to visualise and explore such datasets, especially with the help of colour mapping and other approaches to represent more than the 3 Cartesian dimensions. However, modern software commonly lacks this multi-dimensional functionality or is ineffective. One such limitation is the frequent use of the orthographic projection that produces same-sized points, and this is equivalent to removing one entire dimension.

A novel open-source JavaFX-based plotting framework is presented that focuses on easy exploration of multi-dimensional datasets, and provides unique features and feature combinations to improve knowledge transfer from single stand-alone or interactive plots.

Code metadata

Current code version	1.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00082
Permanent link to Reproducible Capsule	https://philippos.info/repo/philippos/juniper
Legal Code License	GNU Affero General Public License 3.0
Code versioning system used	Git
Software code languages, tools, and services used	Java/JavaFX, FXyz3D, Apache NetBeans IDE
Compilation requirements, operating environments & dependencies	Oracle JDK or OpenJDK (version 21 or newer), Apache Maven
Link to developer documentation/manual	https://github.com/pphilippos/juniper/blob/main/MANUAL.md
Support email for questions	juniper@philippos.info

1. Motivation and significance

Efficiently understanding multi-dimensional data is an important competence in data analytics. Numerous research problems come with many dimensions. A notable case is with design space exploration (DSE) where each resulting data point has a number of attributes associated with it. A subset of them can be treated as optimisation metrics. An example application of DSE is in computer architecture research, where simulations are conducted to estimate the performance and power efficiency of parameterised processor designs.

3D scatterplots can become an excellent communication and exploration tool for such cases, especially if they carry spatial information. Still, when compared to 2D scatterplots, having an extra dimension in space can be a blessing and a curse. It is a blessing because a single 3D scatterplot can give more advanced insights including how each of the dimensions correlate with each other or collectively. For instance, we could produce multiples of 2D plots for a single 3D plot, and would still leave out the interaction between more than a few axes. It is also a curse, however, because in their static form as in scientific manuscripts and computer screens, they need to be flattened to fit in two-dimensional means. This introduces ambiguity.

1.1. Unique visual cues

This paper contributes Juniper, an open-source desktop application specialising in 3D scatterplots and includes a list of unique visual cues. These can be organised according to the corresponding aim:

E-mail address: pp1d24@soton.ac.uk.

https://doi.org/10.1016/j.softx.2025.102203
Received 10 February 2025; Received in revised form 5 May 2025; Accepted 12 May 2025
Available online 31 May 2025

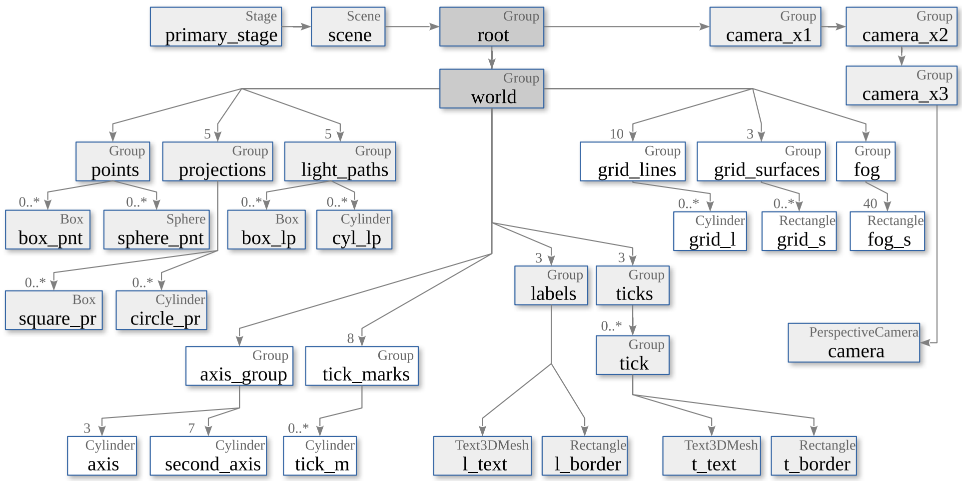

Fig. 1. High-level overview of the class instantiations inside the primary stage of Juniper.

- *Exploration of multiple dimensions*: high number of possible dimensions organised in an association panel, automatic discrete data support including strings, automated point label decluttering such as by applying a top-N selection.
- *Depth perception*: depth-based colour mapping (traditionally only for an axis), fog emulation, a highly adjustable field of view (FoV) applicable to the entirety of the plot.
- *Readability of the plotted points and labels in the 3D space*: multi-axis light paths (alternative to direction drop lines), light paths as 3D objects shaped according to the points (instead of same-sized overlaid lines), 2D projections corresponding to the point shape and colour (not disassociated), 2D surfaces as grids, labels as 3D objects (meshes [1] instead of overlays, while still facing the camera).

1.2. Dependencies and related work

An essential dependency used in Juniper other than JavaFX is FXyz3D, a "JavaFX 3D Visualisation and Component Library" [1]. This allows dynamic creation and manipulation of text meshes in the 3D space. The idea here is for the axis and tick labels to always face the camera, giving the impression of easily-legible overlaid text, while benefiting by having three dimensional attributes, including the label size interacting with the FoV for tick readability.

While a 3D scatterplotting functionality is included in conventional software such as office suites, there are limitations for specialised applications. These relate to the user interactivity, the ability to communicate depth information, as well as to represent multiple dimensions. More feature-rich 3D scatterplots are usually proprietary, costly and closed-source, further restricting their applicability to visualisation research. See Section 4.1 for a more detailed comparison with modern tools.

Juniper's features are partly inspired by existing approaches. Basic point projections on the back of the view box are already used to increase plot readability. One experimental example is a WebGL online 3D scatterplot viewer [2] that features point projections, as well as drop lines whenever the user hovers the mouse cursor over a point. A scatterplot solution that also uses JavaFX 3D and the FXyz3D component library is within the deep space trajectory explorer [3], which specialises in multibody trajectory analysis and is not publicly available. The research-informed ParaProf [4] also uses Java, though it lacks the discussed depth perception and label legibility functionality. With respect to the dimension assignment facility with checkboxes (see the screenshot on the left part of Fig. 2), Juniper is reminiscent of certain software with a limited dimensionality. One such instance is "DB Browser for SQLite", an open source database management system for local databases [5]. It operates on tabular data as well, though it is restricted to 2D plots.

2. Software description

The presented visualisation framework generates easily-readable 3D scatterplots for multi-dimensional data, and is written in JavaFX. As a desktop application, it is entirely controlled by a graphical-user interface (GUI).

2.1. Software architecture

Internally, the core functionality is divided into two Java files corresponding to two windows, one for the controls form, and the one for the interactive 3D plot. The first is mostly a Java Swing code utilising a selection of 2D graphical elements for interfacing with the user, as presented on the left side of Fig. 2. This also has references to JavaFX components to be able to alter the functionality of the other class. The second window has the JavaFX stage where the plot appears, as with the right part of Fig. 2.

Fig. 1 presents an overview of the class instantiations inside the stage class. The primary stage is the window form containing the JavaFX scene, which in turn features the root node. The root node initiates the perspective camera through a series of nested groups for changing the point of view in the 3D space. The world is a child of the root node and contains all physical objects. The leftmost cluster of objects in the diagram is related to the plotted points. There are groups for the point array (either boxes or spheres), and their optional projections and light paths.

The centre cluster of objects relates to the axes organisation, including the axes themselves, secondary (mirror) axes, and FXyz3D text meshes and label borders for both the axes and ticks. The last cluster of nodes relate to grid lines and surfaces, and similar to ticks, they are organised internally into an array or a group of groups, one for each axis. Their cardinality is sometimes more than 3 to facilitate rotation, such as by hiding and unhiding mirrored labels according to which instance is closer to the camera. This hierarchy is essential for manipulating multiple elements with single high-level function calls, such as *setVisble()* from the Node abstract class to hide or show all ticks of an axis.

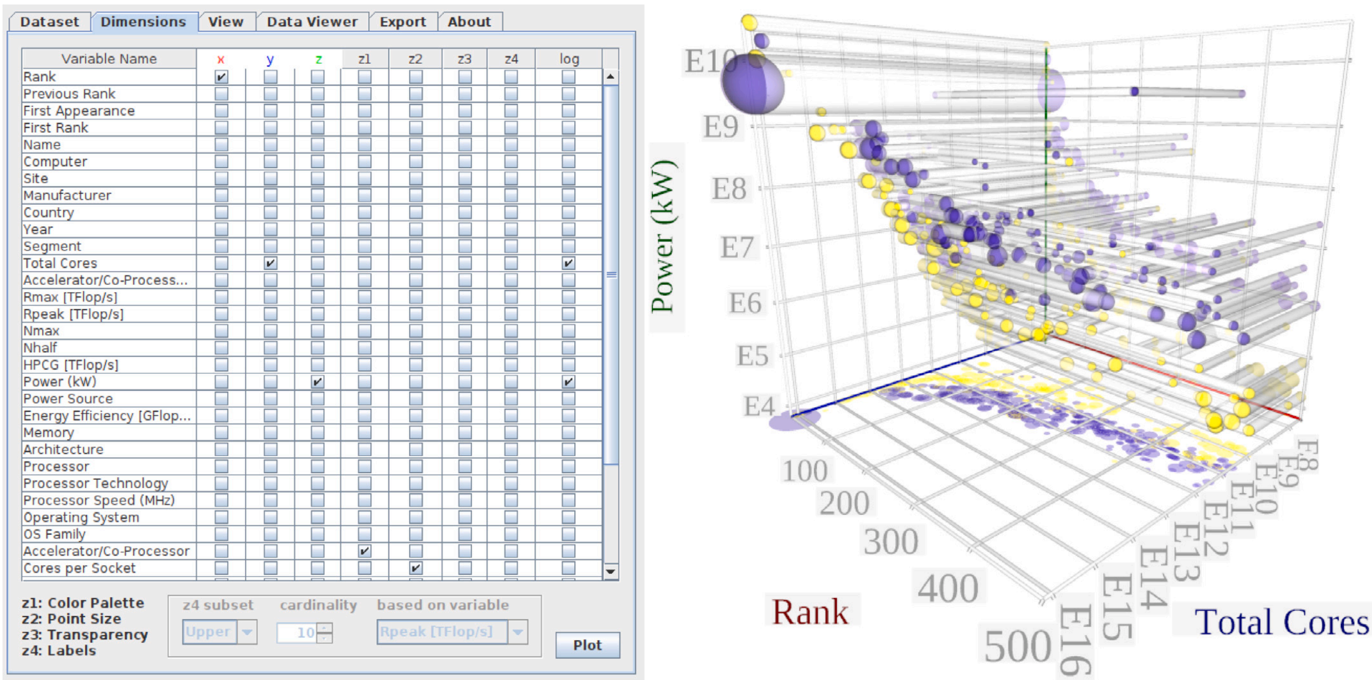

Fig. 2. Generating a 5D plot on top supercomputers by assigning columns to dimensions from a *.csv* file.

2.2. Software functionalities

The main functionality of the software relates to the ability to effectively represent multi-dimensional data. Apart from the 3 dimensions from the Cartesian coordinate system (x, y and z), Juniper also supports colour mapping, point size, point transparency and label overlay (namely *z1* to *z4* respectively). These additional dimensions are naturally supported in a one-to-many relationship between the available variables (column names) from tabular data. This arrangement currently totals 7 dimensions. The example of Fig. 2 shows how the dimension-to-variable assignment is achieved through a table of checkboxes.

Whenever the data format of a dimension is not numerical, a dictionary (hashmap) is automatically built to represent unique strings with an integer, starting with 0 for the first occurrence. Where applicable, the corresponding string appears in each tick mark as opposed to its unique identifier for the first occurrence in the dictionary. This extends the applicability of the framework, such as to still be able to place names on the axes or over the points associated with the data.

One of the highlights of Juniper is a series of features expected to help with depth perception and point readability:

(a) *Field-of-View (FoV):* Exaggerating the FoV can improve the perception of the 3D space in the plot [6] , such as by influencing the object geometry as seen by the camera (observer).
(b) *Depth-based colour mapping:* This feature replaces the $z1$ dimension (colour map) with the values from calculating the distance of each point to the camera (not shown). The colour palette is based on *viridis* [7], which is designed to still be readable after printing in grayscale and by people with colour blindness. The corresponding colour variation in the colour of each point is also applied on their projection, and this is expected to help with associating arbitrary points to their projections.
(c) *Fog:* Similar to the above, fog is designed to help with depth perception as it impacts the visibility of the points (not shown).
(d) *Projections and light paths:* Projections essentially are two-dimensional projections on the surfaces of the view box. They inherit the colour mapping and shape of the points (e.g. circles for spheres), and there is the option to enable them in the xz, yz and xy panes (2-select combinations of the axes). The light paths connect the projection perimeters to the actual points, similar to the umbra in shadows in the real world. Projection and light path examples can be seen in Fig. 2.
(e) *Grid options:* The framework supports two types of grids. First is the line, which is more conventional. Second is the grid surfaces (not shown), which are semi-transparent 3D equivalents of the grid lines, also corresponding to the ticks.

For a detailed description on the usage of the available features with more examples, see the online manual, as specified in .

3. Illustrative example

This section presents an example demonstrating how Juniper can quickly generate insightful stand-alone plots from multi-dimensional data by using a subset of the unique features. The explored dataset is a list of the top 500 supercomputers in the world, which incorporates a comprehensive set of attributes for each supercomputer in a spreadsheet format [8]. The resulting plot is shown on the right side of Fig. 2.

Based on the dimension assignment (see the left side of Fig. 2), the x-axis (red) is the rank of the supercomputer entry. The y-axis (blue) is the total number of CPU (central processing unit) cores in the supercomputer, and the z-axis is the total power in kilowatt-hours. The latter-two have their log checkbox checked for applying a logarithmic scale, due to the high variability in the orders of magnitudes for those values.

There are numerous insights that can be derived by looking at the five-dimensional plot of Fig. 2. First, if we ignore the 4th and 5th dimensions (point size and colour), there seems to be a high correlation between the rank and the total number of cores and power, when we consider them in pairs. If we take the projections on the xy surface (the floor), the more performant (faster) the supercomputer is (rank), the more likely it is to have a higher number of cores. Similarly, the higher the core count, the more likely it is to consume more energy. Additionally, the faster the supercomputer, the more energy it consumes.

Table 1
Comparing related features in modern 3D scatterplot software.

Software name	Tested version	License	FoV adjustability	Drop lines /projections	Internal representation[a]	Main interface[b]	Unique visual cues
Juniper	*1.1 (4/2025)*	GNU AGPLv3	Yes, arbitrary	Yes	3D	GUI	Yes
gnuplot	6.0.2 (12/2024)	gnuplot License	No, orthographic	No	2D	Script	No
GNU Octave [10]	9.4.0 (2024)	GNU GPLv3	No, orthographic	No[c]	2D	Script	No
MATLAB (scatter3)	R2024b Update 5 (2025)	(proprietary)	Yes, 1 increment[d]	No[c]	2D	Script	No
OriginPro	10.1.5.132 (2024)	(proprietary)	Yes, 10 increments	Yes	3D with 2D overlays	GUI	No
SigmaPlot	16.0.0.28 (2024)	(proprietary)	Yes, arbitrary	Yes	2D	GUI	No
AlphaPlot	1.02-stable (1/2022)	GNU GPLv2	Yes, 1 increment	No	3D with raster text	GUI	No
SciDAVis	2.9.2 (3/2022)	GNU GPLv2	Yes, 1 increment	No	2D	GUI	No
PyGMT [11] (Figure.plot3d)	0.3.0-2 (11/2024)	BSD-3-Clause	No, mostly orthographic	No	2D	Script, non-interactive	No

[a] Considered indicative, since the implementation details are not necessarily known for proprietary tools.
[b] Where scripting and graphical components coexist, this is the perceived use case with most features.
[c] Not supported by default, but can be partly emulated with additional scripting.
[d] Alternating between orthographic (default) and realistic perspective [12].

When we take the colour mapping into account (*z1*, Accelerator/Co-processor), we can see that the GPU-equipped supercomputers (yellow) consume less power than CPU-only supercomputers (purple) for the equivalent performance. With the point-size dimension (*z2*, Cores per Socket) we can see that the GPU-equipped supercomputers generally feature smaller CPUs than CPU-only supercomputers. This makes sense due to the fact that CPU-only supercomputers migrate some of the GPU functionality to the CPU sockets, such as by having a high core count (notably the higher ranking ones, which are mostly more recent). The largest sphere in the plot is Sunway TaihuLight in China, the 13th supercomputer in this edition of the list, whose manycore CPUs combine 260 cores in a single socket [9].

With respect to the system behaviour of Juniper as software, navigating the plot of Fig. 2 yields 170 frames per second (FPS) on a Linux machine with an AMD 7950X3D processor, and a memory footprint of 1.38 GB. Indicatively, for a random dataset with a million points and reduced visual cues these values change to 22.7 FPS and 8 GB respectively.

4. Impact

Juniper is targeted towards visualisation research by enabling the exploration of questions relating to visual cues on 3D scatterplots. It is open-source and highly-extensible such as by having the entirety of the plot in the 3D space and only relying on few Java-native dependencies. Hence, Juniper can accelerate research on unconventional scaterplotting techniques, such as the novel visual cues introduced in Section 1.

Ongoing research that uses this framework includes an empirical study exploring the effectiveness of visual cues on 3D scatterplots as static 2D images [6]. This study is conducted on a cohort of 57 participants from an academic community and studies a subset of the aforementioned unique visual ques. It distinguishes certain combinations of visual cues as the most effective with an up to 5.2x and 1.7x error and time reduction respectively when reading data points from a conventional 3D scatterplot.

More widely, as an advanced 3D scatterplotting tool, Juniper can be used to conduct research that relies on knowledge extraction from complex data. With its innovative features, it focuses on quickly and effectively expressing multiple-dimensions with realistic 3D scatterplots. In combination with its graphical user interface, it can provide early insights from experimental data with high dimensionality.

4.1. Comparison with alternatives

Table 1 introduces a comparison of Juniper with prominent frameworks capable of generating 3D scatterplots. To the best of the author's knowledge, all competing solutions lack the unique visual cues as presented in Section 1.1. Hence, this discussion is limited to related aspects.

OriginPro [13] could be considered the closest framework in regard to the explored functionality, and supports a superset of plot types. On 3D scatterplots, a mix of 2D raster renderings and vector [11] components are used to create an impressive clarity effect in printouts (axis labels in a text format). This results in some loss of physical properties and interactive navigation flexibility, such as with quantised ranges for rotation and FoV adjustment, and some pixelation when zooming in. Being a more general and proprietary analytics framework, this is reflected in the license cost at USD 645 for a node-locked annual subscription.

Similarly, MATLAB R2024b costs USD 940 per year for an individual License, but relies heavily on additional scripting and add-ons for non-basic plots. Existing open-source counterparts can provide more flexibility with their openness, but they often inherit or introduce limitations. These include GNU Octave [10], which does not support all MATLAB functions of interest [12], and AlphaPlot, whose 3D scatterplot labels are currently rendered as low-resolution textures and are less legible than OriginPro.

Keeping intuitiveness in mind, it could still be possible to find more resemblance in related work, or be able to reproduce the unique features with additional scripting. In this aspect, the contribution is at least their coexistence in Juniper, while being open-source and research-friendly.

5. Conclusions

A novel open-source framework is presented for plotting multi-dimensional 3D scatterplots. It provides point-and-click functionality as a Java-based desktop application for quickly applying advanced visual cues and additional dimensions to 3D scatterplots. A series of unique features relate to the easy association and representation of multiple dimensions from tabular data, as well as the increase of depth perception. The framework is indicatively demonstrated with the use of a publicly-available dataset on high-performance computing , where multiple advanced and specialised insights are drawn from a single multi-dimensional plot.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Early high-level editorial feedback from Dr Lucy Hederman is gratefully appreciated. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] F(X)yz. FXyz3D: A JavaFX 3D visualization and component library. 2013, URL https://github.com/FXyz/FXyz. [Accessed September 2024].

[2] Hergott M. 3D scatter plot and spatial visualization. 2013, URL https://miabellaai.net/, [Accessed August 2024].

[3] Phillips SM, Davis DC, Sweeney DJ. Cloud computing methods for near rectilinear halo orbit trajectory design. In: AAS/AIAA astrodynamics specialists conference. (AAS 19-874). 2019.

[4] Bell R, Malony AD, Shende S. Paraprof: A portable, extensible, and scalable tool for parallel performance profile analysis. In: Euro-par 2003 parallel processing: 9th int. euro-par conf.. 2003, p. 17–26. http://dx.doi.org/10.1007/978-3-540-45209-6_7.

[5] Piacentini M, et al. DB Browser for SQLite. 2024, URL https://sqlitebrowser.org/. [Accessed December 2024].

[6] Papaphilippou P, Hederman L. Empirical study on the representation of 3D scatterplots as 2D figures. 2025, arXiv:2406.06146.

[7] Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Pedro A, Sciaini, Marco, Scherer, Cédric. viridis(Lite) - Colorblind-Friendly Color Maps for R. 2024, http://dx.doi.org/10.5281/zenodo.4679423, viridis package version 0.6.5URL https://sjmgarnier.github.io/viridis/.

[8] TOP500org. June 2024 List. TOP500 63rd Ed 2024. URL https://www.top500.org/lists/top500/2024/06/.

[9] Fu H, Liao J, Yang J, Wang L, Song Z, Huang X, Yang C, Xue W, Liu F, Qiao F, et al. The Sunway TaihuLight supercomputer: system and applications. Sci China Inf Sci 2016;59:1–16. http://dx.doi.org/10.1007/s11432-016-5588-7.

[10] Eaton JW, Bateman D, Hauberg S, Wehbring R. GNU Octave version 9.4.0 manual: a high-level interactive language for numerical computations. 2024, URL https://www.gnu.org/software/octave/doc/v9.4.0/.

[11] Uieda L, Tian D, Leong WJ, Toney L, Newton T, Wessel P. PyGMT: A Python interface for the Generic Mapping Tools. 2020, http://dx.doi.org/10.5281/zenodo.4253459.

[12] MathWorks. Understanding view projections. 2025, MATLAB Help Center URL https://uk.mathworks.com/help/matlab/creating_plots/understanding-view-projections.html. [Accessed April 2025].

[13] OriginLab Corp. Origin: Data analysis and graphing software. 2024, URL https://originlab.com/index.aspx?go=Products%2FOrigin. [Accessed 10 2024].