The Journal of Open Source Software

Regl-Scatterplot: A Scalable Interactive JavaScript-based Scatter Plot Library

Fritz Lekschas [1]

1 Independent Researcher, USA

DOI: 10.21105/joss.05275

Software

- Review
- Repository
- Archive

Editor: Fabian Scheipl

Reviewers:

- @Fil
- @xiaohk

Submitted: 15 February 2023
Published: 11 April 2023

Summary

Scatter plots are one of the most popular visualization methods to surface correlations (like trends or clusters) in bivariate data. They are used across all scientific domains. With datasets ever increasing in size, it can become challenging to effectively render and explore scatter plots. Hence, there is a need for scalable and interactive scatter plot libraries.

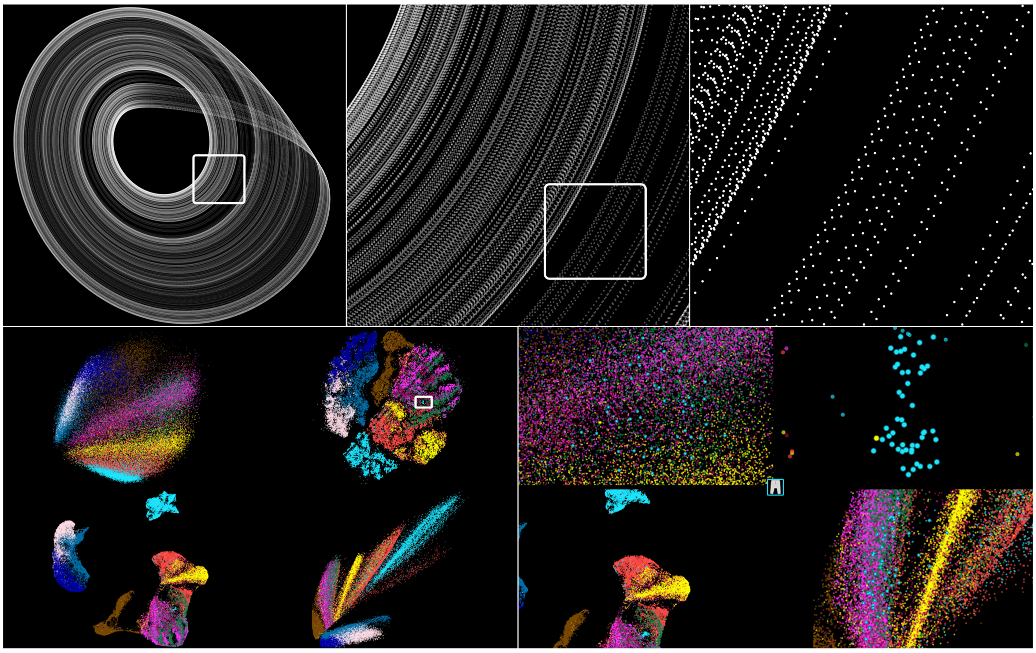

Figure 1: Examples of `regl-scatterplot`. The top row visualizes the Rössler attractor and demonstrates the dynamic point opacity. As one zooms in (see the white bounding box) and number of points in the view decreases, the point opacity increases. The bottom row shows four embeddings of the Fashion MNIST dataset (Xiao et al., 2017) created with PCA (Pearson, 1901), t-SNE (Maaten & Hinton, 2008), UMAP (McInnes et al., 2018), and a variational autoencoder (Kingma & Welling, 2013). The embeddings are visualized using four `regl-scatterplot` instances that synchronize and zoom to the selected points. For instance, upon selecting the sky blue cluster of points in the top-right instance (see the white bounding box), the same points are selected in the other instances. Also, all four instances zoom to the selected points (right side of the second row).

`regl-scatterplot` is a general-purpose data visualization library written in JavaScript for rendering large-scale scatter plots on the web (Figure 1). Every aspect of the library focuses on performance. Thanks to its WebGL-based renderer, which is written with `regl` (Lysenko, 2016), the library can draw up to twenty million points while offering smooth pan and zoom.

To interact with the data points, `regl-scatterplot` implements fast lasso selections using a spatial index (Agafonkin, 2016).

Beyond the rendering and interaction performance, visualizing large datasets as scatter plots also poses perceptual challenges (Micallef et al., 2017). In particular, the right level of opacity is critical to faithfully represent the data distribution while ensuring that outliers are perceivable. To simplify this aspect of the scatter plot design, `regl-scatterplot` implements a density-based point opacity that extends an approach by Reusser (2022). In addition to the original approach, the opacity dynamically adjusts to the points within the field of view as the user pans and zooms. Finally, `regl-scatterplot` supports drawing spline-interpolated point connections and animated transitions of the point location and color encoding when drawing a new dataset with point correspondences (Figure 2).

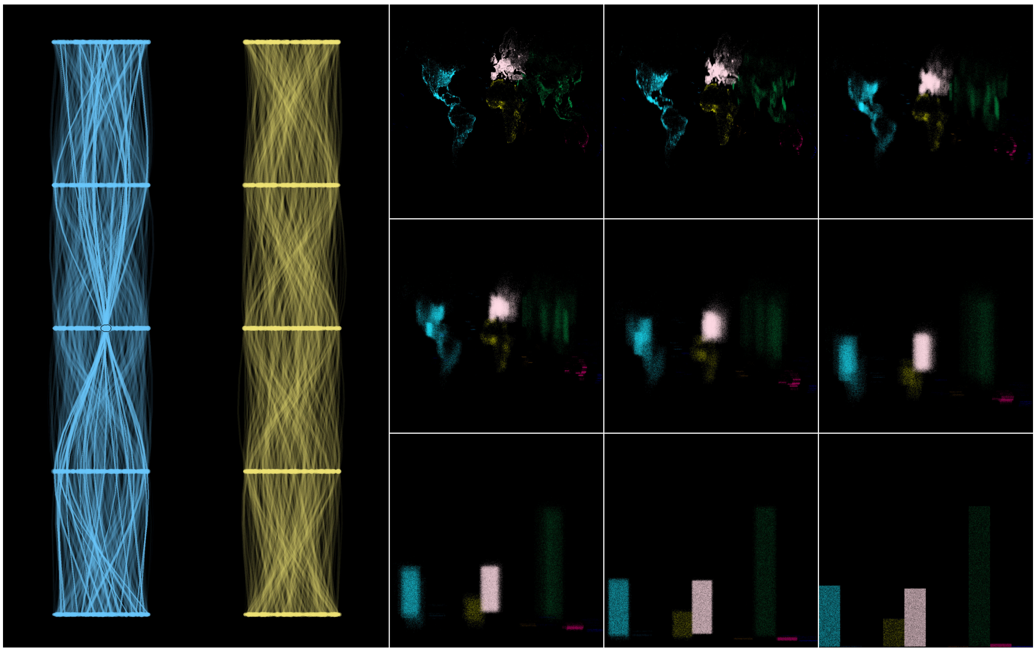

Figure 2: Additional features of `regl-scatterplot`. On the left side, we show an example of point connections rendered as spline-interpolated lines. Note that connections of selected points are highlighted as well. On the right side, we show nine key frames of an animated transition of the point locations from a geographical to a bar chart representation. The points visualize cities across the globe (GeoNames, 2023). The animation example was inspired by Beshai (2017).

Statement of Need

`regl-scatterplot` was designed for visualization researchers and practitioners. It has already been cited in a number of scientific publications (Bäuerle et al., 2022; Lekschas et al., 2020; Narechania et al., 2022; Santala, 2020; Warchol et al., 2023), it formed the software foundation for a computer science master thesis (Hindersson, 2021), and it is actively used in scientific software tools (Cabrera et al., 2023; Cube, 2022; Lekschas, 2020, 2022; Li, 2022; Rau, 2021; Warchol et al., 2023). The focus on scalable rendering and interactions in combinations with a wide variety of design customizations in `regl-scatterplot` enables visualization researchers and practitioners to build, study, and test new visualization tools and applications for scalable exploration of the ever-increasing number of large-scale datasets.

Related & Future Work

There are several related JavaScript packages for rendering scatter plots on the web. General-purpose visualization libraries like `d3` (Bostock et al., 2011) or `vega-lite` (Satyanarayan et al., 2016) are broadly useful, but are not well suited to render datasets with millions of data points due to the reliance on a Document Object Model (in the case of `d3`) and limited support for GPU-based rendering. Like `regl-scatterplot`, `CandyGraph` (Terrell, 2023) overcomes this limitation by using WebGL for the rendering. However, being a general-purpose plotting library means that `CandyGraph` does not offer critical features for exploring scatter plots interactively like pan-and-zoom, lasso selections, etc. The visualization charting library `plotly.js` (Plotly, Inc, 2023) has support for WebGL rendering and offers interactive pan-or-zoom and lasso selection, but is lacking other features like animated transitions, dynamic point opacity, or synchronization of multiple scatter plot instances. Similarly, the bespoke `regl-scatter2d` (Ivanov, 2023) library offers scalable WebGL-based rendering of scatter plots and allows customizing the point shape. However, it does not support data-driven visual encodings or interactive point selections. Finally, `deepscatter` (Schmidt, 2023) is another scalable scatter plot library that offers data-driven visual encodings. Its reliance on tile-based data enables even greater scalability compared to `regl-scatterplot` but at the expense of having to preprocess data. Also, as of today, the library is lacking support for lasso selections or the ability to synchronize multiple scatter plot instances.

In the future, we plan to add built-in support for streaming tiled Apache Arrow files in `regl-scatterplot` to further improve the performance. We also plan to move as much or ideally all of `regl-scatterplot`'s code to web workers, which would help to avoid any performance impact on the main thread.

Acknowledgements

We acknowledge contributions from Jeremy A. Prescott, Trevor Manz, and Emlyn Clay, and Dušan Josipović.

References

Agafonkin, V. (2016). *Regl: Functional WebGL*. https://github.com/mourner/kdbush

Bäuerle, A., Cabrera, Á. A., Hohman, F., Maher, M., Koski, D., Suau, X., Barik, T., & Moritz, D. (2022). *Symphony: Composing interactive interfaces for machine learning*. https://doi.org/10.1145/3491102.3502102

Beshai, P. (2017). *Beautifully animate points with WebGL and regl*. https://peterbeshai.com/blog/2017-05-26-beautifully-animate-points-with-webgl-and-regl/

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Cabrera, Á. A., Fu, E., Bertucci, D., Holstein, K., Talwalkar, A., Hong, J. I., & Perer, A. (2023). Zeno: An interactive framework for behavioral evaluation of machine learning. *CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3544548.3581268

Cube, E. D. (2022). *EODash: Software behind the RACE dashboard by ESA and the European Commission as well as the Earth Observing Dashboard by NASA, ESA, and JAXA*. https://github.com/eurodatacube/eodash

GeoNames. (2023). *GeoNames*. https://www.geonames.org

Hindersson, T. (2021). *Scatterplot patterns: Using animation to enhance scatterplots* [Master's thesis, KTH, School of Electrical Engineering; Computer Science]. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-305905

Ivanov, D. (2023). *Regl-Scatter2d: Fast and precise scatter plot*. https://github.com/gl-vis/regl-scatter2d

Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational Bayes*. arXiv. https://doi.org/10.48550/ARXIV.1312.6114

Lekschas, F. (2020). *Peax: A visual pattern explorer for epigenomic data using unsupervised deep representation learning*. https://github.com/Novartis/peax

Lekschas, F. (2022). *Jupyter-Scatter: An interactive scatter plot widget for Jupyter Notebook, Lab, and Google Colab*. https://github.com/flekschas/jupyter-scatter

Lekschas, F., Peterson, B., Haehn, D., Ma, E., Gehlenborg, N., & Pfister, H. (2020). Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. *Computer Graphics Forum*, *39*(3), 167–179. https://doi.org/10.1111/cgf.13971

Li, G. (2022). *GoTreeScape prototype system*. https://github.com/bitvis2021/GoTreeScape

Lysenko, M. (2016). *Regl: Functional WebGL*. https://github.com/regl-project/regl

Maaten, L. van der, & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform manifold approximation and projection for dimension reduction*. arXiv. https://doi.org/10.48550/ARXIV.1802.03426

Micallef, L., Palmas, G., Oulasvirta, A., & Weinkauf, T. (2017). Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, *23*(6), 1588–1599. https://doi.org/10.1109/TVCG.2017.2674978

Narechania, A., Karduni, A., Wesslen, R., & Wall, E. (2022). VITALITY: Promoting serendipitous discovery of academic literature with transformers & visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, *28*(01), 486–496. https://doi.org/10.1109/TVCG.2021.3114820

Pearson, K. (1901). On lines and planes of closest fit to systems of points is space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Plotly, Inc. (2023). *Plotly.js: Open-source JavaScript charting library behind Plotly and Dash*. https://github.com/plotly/plotly.js

Rau, A. (2021). *histoCAT web: Histology topography cytometry analysis toolbox*. https://github.com/BodenmillerGroup/histocat-web

Reusser, R. (2022). *Selecting the right opacity for 2D point clouds*. https://observablehq.com/@rreusser/selecting-the-right-opacity-for-2d-point-clouds

Santala, S. (2020). *Fast interactive design of scatterplots for large data set visualisation*. 1–6. https://doi.org/10.1145/3334480.3381443

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2016). Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 341–350. https://doi.org/10.1109/TVCG.2016.2599030

Schmidt, B. (2023). *Deepscatter: Zoomable, animated scatterplots in the browser that scales over a billion points*. https://github.com/nomic-ai/deepscatter

Terrell, R. (2023). *CandyGraph: A flexible and fast-by-default 2D plotting library tuned for rendering huge datasets on the GPU at interactive speeds*. https://github.com/wwwtyro/candygraph

Warchol, S., Krueger, R., Nirmal, A. J., Gaglia, G., Jessup, J., Ritch, C. C., Hoffer, J., Muhlich, J., Burger, M. L., Jacks, T., Santagata, S., Sorger, P. K., & Pfister, H. (2023). Visinity: Visual spatial neighborhood analysis for multiplexed tissue imaging data. *IEEE Transactions on Visualization and Computer Graphics*, *29*(1), 106–116. https://doi.org/10.1109/TVCG.2022.3209378

Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms*. arXiv. https://doi.org/10.48550/ARXIV.1708.07747