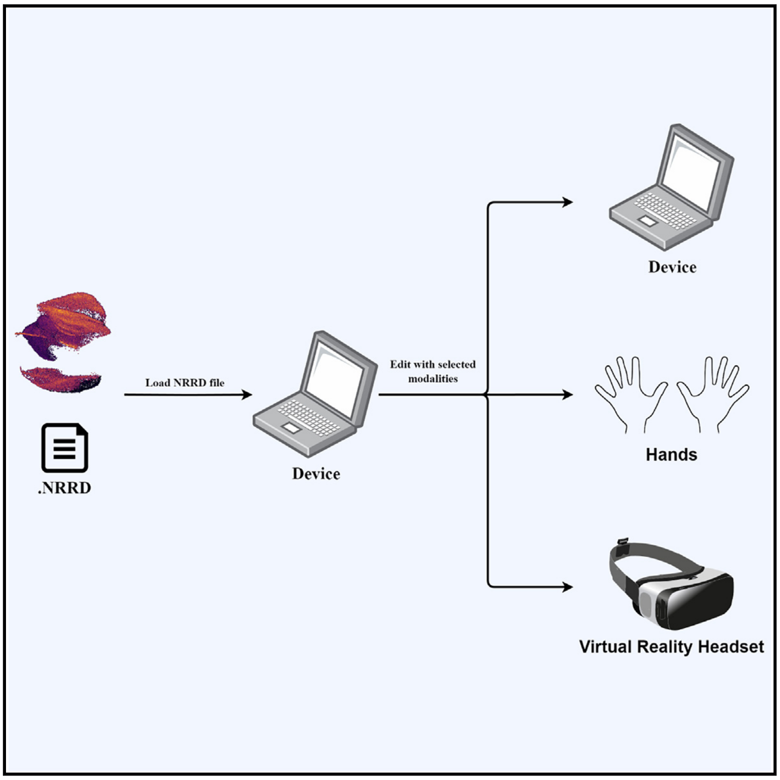
iScience

Article

QuickVol: A lightweight browser tool for immersive visualizations of volumetric data

Graphical abstract



Highlights

- Developed platform for visualizing volumetric NRRD (Nearly Raw Raster Data) files
- Visualization system designed to run in any modern web browser, can be used offline
- Built for efficiency, works on mobile devices and slow internet connections
- View data with a camera-only hand tracking mode or immersive virtual reality mode

Authors

Maxim Kuznetsov, Mircea Teodorescu, Mohammed A. Mostajo-Radji, Sri Kurniawan

Correspondence

maakuzne@ucsc.edu

In brief

Applied sciences; Computer science; Software engineering

Kuznetsov et al., 2024, iScience *27*, 111379
December 20, 2024
https://doi.org/10.1016/j.isci.2024.111379

CellPress

iScience

CellPress
OPEN ACCESS

Article

QuickVol: A lightweight browser tool for immersive visualizations of volumetric data

Maxim Kuznetsov,[1,2,4,*] Mircea Teodorescu,[2,3] Mohammed A. Mostajo-Radji,[2] and Sri Kurniawan[1,2]
[1]Department of Computational Media, University of California Santa Cruz, Santa Cruz, CA, USA
[2]Genomics Institute, University of California Santa Cruz, Santa Cruz, CA, USA
[3]Department of Electrical and Computer Engineering, University of California Santa Cruz, Santa Cruz, CA, USA
[4]Lead contact
*Correspondence: maakuzne@ucsc.edu
https://doi.org/10.1016/j.isci.2024.111379

SUMMARY

Volumetric layouts of data are becoming increasingly common in a number of fields. Visualizing these data often requires downloading a large suite of dedicated tools with a significant learning curve. This process can be overwhelming for students or new researchers looking to quickly visualize and showcase a volumetric dataset. QuickVol was developed as a system to allow for rapid viewing of volumetric data without requiring extra setup. Built on WebGL, our system can run on any modern web browser, including mobile browsers, and can work completely offline. Additionally, an experimental immersive hand-tracking feature is included, which allows for hands-free manipulation of the imported volume, along with a showcase mode for viewing with a virtual reality headset.

INTRODUCTION

Volumetric data are commonly captured in a wide range of fields, especially in medical and biological imaging. As the data captures values in three-dimensional (3D) space, its use is important in visualizing information where a flat projection may be insufficient. By providing a 3D matrix of data points, volumetric data allow one to "see through" volumes that may be opaque from an outside perspective. Frequently, this feature of volumes is used to analyze the internal structure of an object without perturbing the object itself.

A number of tools have been built to analyze this type of data format, with various strengths and weaknesses. A majority of these applications require a desktop computer to operate.[1–4] The dedicated platforms are powerful as they include a large collection of tools and methods that can assist in advanced analysis of the dataset. However, in a new user who has no experience in using tools for volumetric analysis, the complexity of the tooling presents a significant learning curve. Additionally, some of the more feature-rich applications are proprietary and require a large upfront cost, which can be prohibitive to users without institutional support.

Although processing and displaying these volumetric images can be computationally expensive, advances in computing power have allowed real-time rendering of volumes to be feasible. In the same time frame, consumer virtual reality headsets have entered the market at a reasonable price point. These newer headsets, such as the Meta Quest 3, incorporate methods of human-computer interaction (HCI) that are unique in comparison to the standard flat screen with a mouse and keyboard. Due to the public reach of new HCI methods offered by consumer headsets, it has been an area of active theoretical and experimental research.[5] One of the experimental methods of input offered by headsets such as the Meta Quest 2 (and newer headsets) is optical tracking of hand positions, without any extra tracking devices. Using hands-free interaction, users are capable of manipulating scenes with natural hand movements. A number of tools and experiences[6,7] have started to explore natural hand input as the primary mode of interaction over traditional input devices.

Several platforms have been released that have a specific focus on volumetric rendering in the web browser. The most modern solutions are Kitware's VolView[8] and Glance,[9] both of which are based on vtk.js.[10] Both run directly in the web browser, without any requirement for the user to run a local dedicated instance outside of the browser. VolView is a more feature-rich renderer compared to Glance, as it supports multiple render modes and extra features such as annotation capabilities. However, VolView focuses solely on radiological data and as such only supports loading DICOM files. Glance is intended to be a lightweight version of Kitware's ParaView,[11] acting as a general volume renderer that does not focus on a single use case. Unlike VolView, Glance does not incorporate the more advanced rendering modes and annotation features. Notably, although both VolView and Glance are built on vtk.js, which has support for WebXR features, neither display capability of interacting with WebXR in any capacity.

Another web platform for visualizing volumetric data is First Person Bioimage (FPBioimage).[12] It was one of the first solutions built specifically for sharing and rendering volumetric data in the web. FPBioimage enables users to host their own instance that directly loads the volume as you open the page, which can be useful in presentation contexts. Additionally, FPBioimage

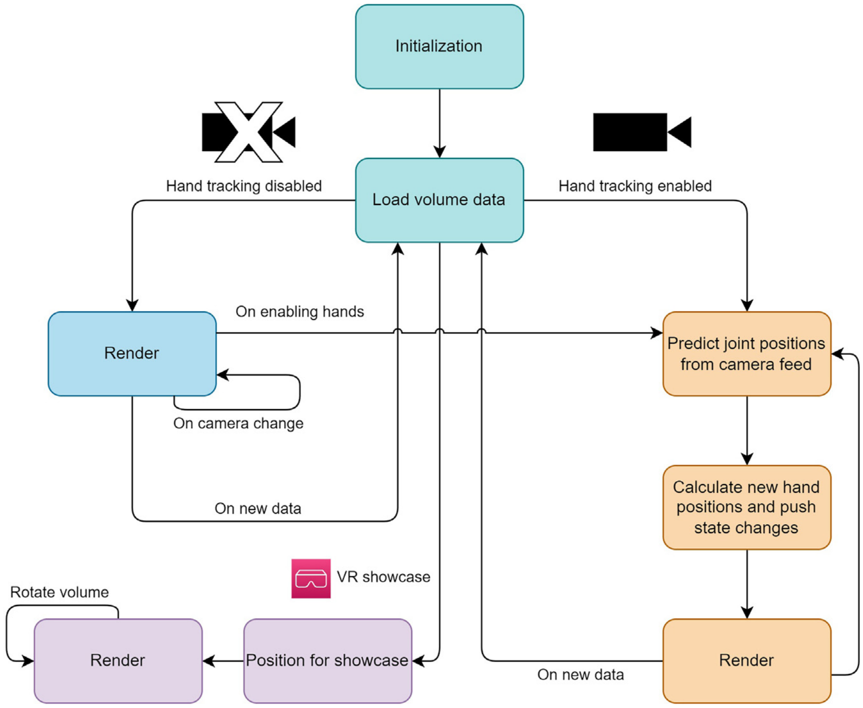


Figure 1. State diagram for usage with hands tracking enabled or disabled

supports viewing the volumetric data using a virtual reality headset through the mobile app (this feature is not available in the web version). Compared to VolView and Glance, FPBioimage offers a more stripped down set of features, with an additional restriction of requiring the volume to be loaded as separate PNG files. Along with sequentially loading each layer, the initial loading time is further increased due to FPBioimage running on a WebGL version of Unity, which is primarily intended to be used as a video game engine, rather than a slimmed down visualization platform. FPBioimage also requires users to host their own instance of the program to visualize files, as it does not support uploading files from the browser.

Both Kitware's solutions and FPBioimage have advantages and disadvantages that would have to be considered for the particular use case of the user. Neither have explored the recent forms of XR interaction, such as hands-free interaction with hand tracking. Kitware's solution can technically access this through vtk.js, but the current versions do not expose these features, and anyone wanting access to WebXR would have to implement it manually.

In developing our tool, we sought to create a low barrier of entry in visualizing volumetric data, particularly data that were being created and shared in our local research groups. To share interactive data from new developments, one was required to download a suite of tools and learn their interfaces to simply view the data. For an interested member or passerby who may not be working with volumetric data regularly, the setup process required for interacting with the data may prevent some from having more interest. Therefore, we developed QuickVol as a way to interact with the volumetric data with minimized setup steps. The user is only required to go to a small web page, where they can then select the file locally (through a file picker or drag and drop) to render. Features for customizing the rendering style and performance have been added, but they are kept purposely minimal to not overwhelm a first time user with the interface. Additionally, with the rise of recent interest in virtual and augmented reality interfaces such as the Oculus Quest 3, we have included a control system that allows users to rotate and scale the volume using only their hands. Unlike pure WebXR systems, our implementation does not require hand tracking devices, as it derives hand positions from a single camera. A guide detailing each interaction mode in the tool can be watched in Video S1. Photos of the immersive modes can be viewed in Figure S3.

We prototyped QuickVol in two different types of 3D biological data: first, we used open-source 3D images, as well as newly generated light-sheet microscopy image stacks of brain organoids. Then, we demonstrated its application in mining single-cell RNA sequencing data.[13–15] Together, we showed that QuickVol is a versatile, light-weight, and easy to use tool with direct applications in biomedical data.

RESULTS

System framework

As a platform modeled with the idea of quick and frictionless sharing of visualizations, the cost of the user loading the framework must to be minimal. We chose to build the platform using Three.js,[16] as it provides a light wrapper around WebGL calls while also allowing for rapid prototyping and simple distribution. Three.js produces a small final build, which is beneficial in regions where internet connections are slow or unstable, as the total script size fetched from the server is low. This is in contrast to Unity, another framework used to create rapid prototypes of interactive visualizations, which offers a more robust suite of development tools, but requires a much larger bundle to be downloaded and has a runtime that is not developed with visualization in mind.

Rendering methods

The method used to render the volumetric data is based on the volumetric renderer included within Three.js. It is run on the GPU through the fragment shader, which uses volume ray casting to display the volume with either extracted isosurfaces or a maximum intensity projection (MIP). The shader included in Three.js to accomplish this does not support matrix transformations applied to the volumetric data. Therefore, the shader was

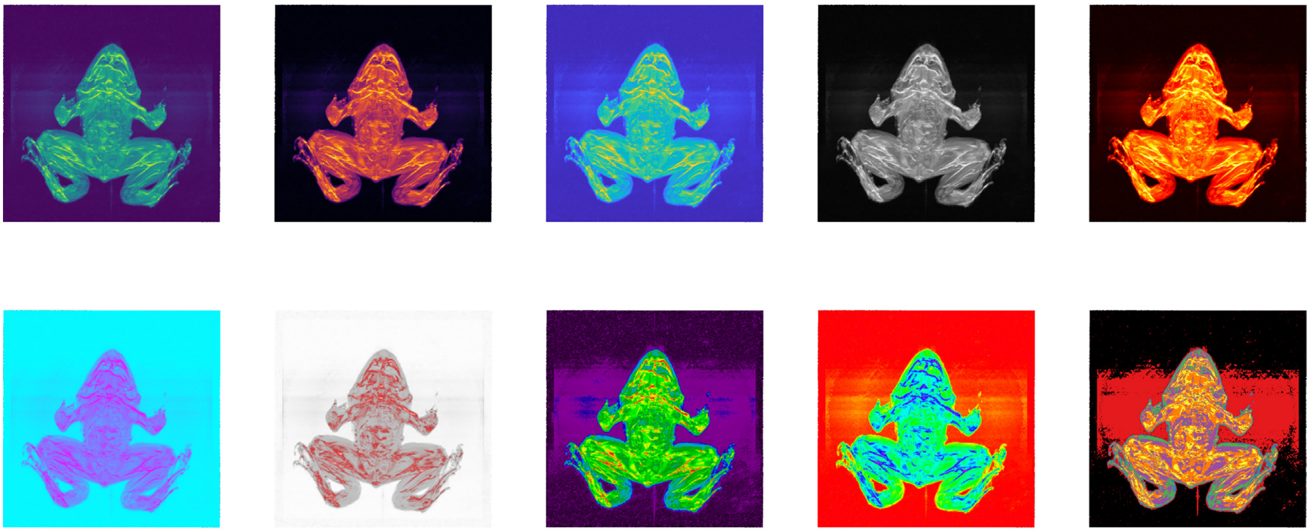


Figure 2. A selection of the available colormaps applied to a frog MRI scan displayed using MIP

The top row features widely used colormaps, whereas the bottom row features more uncommon or niche colormaps.

rewritten, using the base Three.js volume shader as a reference point but with added support to be able to translate, rotate, and scale the volume independent of the camera transform. A high-level state diagram of the interaction and rendering pipeline for QuickVol can be seen in Figure 1.

MIP rendering consists of finding the final color of each pixel based on the maximum value found in each ray trace. This rendering style is fast to calculate and can be adequate in projecting a picture of the contents within the volumetric data. For this reason, it is set as the default rendering mode in QuickVol. Isosurface rendering is done using the Blinn-Phong shading model.[17] A ray is traced until the minimum threshold value set by the user is exceeded. After determining the value of the final traced voxel, illumination of the pixel from the trace is calculated as follows:

$$I = k_a i_a + \sum_{m}^{lights} k_d i_{m,d} (N \cdot L) + k_s i_{m,s} \left[\left(\frac{L+V}{\|L+V\|} \right) \cdot N \right]^{\alpha}$$

In this illumination formula, k_a, k_d, k_s are the intensities of the ambient, diffuse, and specular components; $i_a, i_{m,d}, i_{m,s}$ are the colors of the ambient, diffuse, and specular components; N, L, V are the surface normal, light direction vector, and view direction vector; and α is the shininess of the specular component. The constants from the formula used in QuickVol were set to values that would display each of the test volumes in a visually appealing way; however, these constants in the fragment shader can easily be modified by the user to suit their need. Notably, the isosurface rendering style is not set to default as it requires more computation power in comparison to MIP rendering. However, it can generate a more visually impressive image by simulating lighting on 3D surfaces. This allows for possibly greater clarity in the visualization as the user can select for structures to visualize within the volume data, rather than projecting through the entire volume.

As there can be wide variations in how the input data are distributed, an array of colormaps is offered. The range of the colormap being used can be selected by the user for further customization of the final visual output. Currently, only a static set of colormaps is offered; however, a majority of the most commonly used colormaps are offered. A set of the colormaps can be seen in Figure 2.

Similarly, the platform also offers users the option to select the maximum number of z stacks being rendered from the dataset. As the microscopy data that was compiled before the development of QuickVol was taken by offsetting the z axis, it was important for the research team interested in that data to select the top z stack being displayed. With two distinct rendering modes, a number of colormaps, and max z stack selection, QuickVol can cover a range of basic visualization configuration that are required for most sets of volumetric data.

Rendering detail

As a number of users could be accessing the platform from low-power devices, options modifying the detail level of the volume being rendered are necessary for interactive usage. Older mobile phones or laptops, for example, may struggle with keeping a usable framerate when rotating, translating, or scaling a large volume. Therefore, we have included two methods to reduce the quality of the output image in the case of interactive usage. The first option is the ability to modify the size of the step each ray takes through the image. While setting this parameter too high can quickly degrade the image, this option can be used to find the proper orientation of a volume for analysis before increasing detail. The second option will discard a set number of rays, reducing the computation required to render the image to the screen. Similarly to the first option, setting this value too high can degrade the final output, but a general view of the data can still be seen even with this parameter set to a high value.

Under the default parameters, the volume is rendered on demand, meaning that no extra computation will take place when

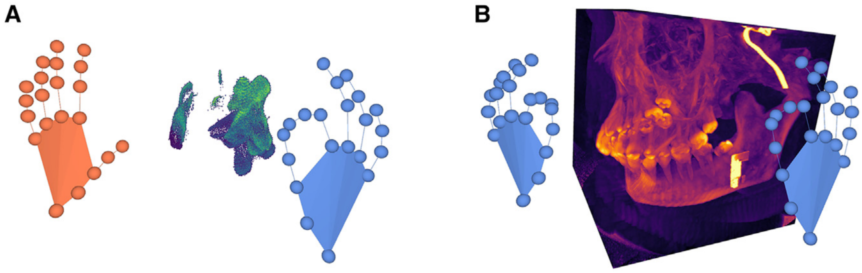


Figure 3. Visual feedback of user interacting with volume
(A) Right hand being used for the rotation command.
(B) Both hands being used for the zoom command.

there have been no inputs to translate, rotate, or scale the volume. For low-power devices, rendering detail can be turned down to enable fast interactivity with modifying volumes and then brought back up to full to take screenshots of the data or to obtain a clearer look of more intricate parts of the dataset. On-demand rendering also enables low power usage if the volume is not being modified, which is especially important for mobile devices or in showcases where the platform may be visible for extended periods of time. It is important to note that with hand tracking enabled and actively tracking (discussed in the next subsection), there is essentially constant re-rendering, as the hand positions are always changing. This requires greater power usage in comparison to on-demand rendering without hand tracking and should be taken into consideration if running QuickVol off of a device using battery power.

Hand tracking

Input for the hand positions is processed using the MediaPipe library.[18] MediaPipe uses computer vision machine learning models to predict features such as body pose and hand position using only a single camera feed. It is developed to be optimized specifically for mobile devices and the web, capable of producing usable results running locally in real time. The hand tracking model supplies hand landmark positions, including a prediction of the depth of the position. The model is moderately sized and downloaded to the client once the user initiates the hand tracking request. It can also be downloaded and run locally, negating the need for an active internet connection during initialization.

A possible option considered for hand tracking in a web browser was the WebXR Hand Input API.[19] This API is part of the larger WebXR standard, which is supported by the majority of modern browsers. The Hand Input API provides hand positions in a standardized format that hand trackers can implement. However, use of this API requires a device that can supply hand positions through this standard, and currently the number of devices that support this standard is low (most notable examples being the Meta Quest 2 and 3). Additionally, hand prediction with a simple camera input is not possible, unless the user is using an emulator for the API.[20] Due to these reasons, we decided to forgo usage of the WebXR Hand Input API, resulting in wider user support from the single camera input option.

Locomotion

QuickVol allows for full rotation, translation, and scaling of the loaded volume. These actions can be controlled using a mouse with a left click, right click, or the scroll wheel, respectively. On touchscreen devices, multi-touch input is supported, which enables all three forms of transformations. Finally, in the case of hand tracking usage, only rotation and scaling are supported. Due to the current inaccuracies in hand position prediction, we were restricted to choosing a small set of actions that are crucial for interactivity. Therefore, we have restricted hand input to only rotating and scaling the object. The rotation mode is activated by pinching the right thumb and right index finger, and the scaling mode is activated by pinching the thumb and index finger on both hands. Visual feedback given to the user during hand input locomotion can be seen in Figure 3. The two hand actions provide an adequate level of interactivity while still being usable under conditions where hand prediction may be unstable. The set of transformations available between the device and hand tracking use case can be seen in Figure 4.

QuickVol offers an immersive virtual reality (VR) view of the scene, which can be used to showcase the rendered volume. This feature is only available if a VR output method is detected. For this, the user is required to connect a device that supports WebXR and permits their web browser to use VR devices. In the immersive mode, all forms of locomotion are disabled, and the user is placed at a fixed point relative to the volume. The volume then is rotated in a semi-random but smooth style to give the user coverage of all sides of the volume over time. While in the immersive VR mode, the user is capable of moving their head with six degrees of freedom; however, the scale of the volume being rendered is not set small enough that a user would be able to fully walk around it. In this mode, hand tracking is disabled due to the possible incongruence with the camera angle used for tracking hands and the headset position.

Supported formats

In the current iteration, the system only accepts Nearly Raw Raster Data (NRRD) files as the input for visualizing volumetric data. Therefore, to render any data, it must first be converted into the NRRD format before being uploaded into the system. The NRRD format was chosen due to the simplicity and level of support of the data format.[21] Users can use programs such as Fiji[1] to convert their volumes to the NRRD format or create custom workflows for more complex data if necessary. As QuickVol is currently only a viewer, there are no output formats, other than scene screenshots that are saved as a PNG.

Initial development was conducted incrementally with a multidisciplinary panel of subject experts. Feedback and iteration were completed with parties involved in biological visualizations and HCI system design. After a large portion of the system development was completed, informal feedback was gathered

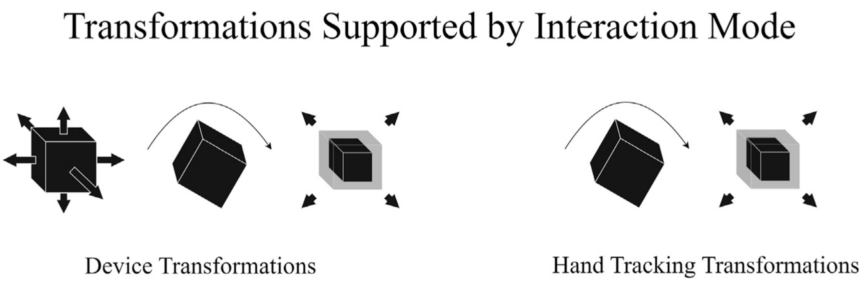


Figure 4. Types of transformations supported by usage mode

Standard device usage allows for translation, rotation, and scaling. Hand tracking allows for rotation and scaling. The virtual reality showcase mode is not listed, as it does not support transformations.

from subjects who were not familiar with either fields to assess usability. The first version of the experimental prototype was completed after considering and incorporating responses from non-expert subjects.

User study

System usability was experimentally evaluated with subjects who did not have previous experience in volumetric visualization and analysis. A total of 9 groups of subjects were recruited for the feasibility study, which was approved by the University of California Santa Cruz Institutional Review Board (HS-FY2021-66). Each group had one to four individual participants. At the start of the procedure, a brief introduction ($t < 1$ min) was given on the data being displayed and the hand gesture actions available. At this point in time, the prototype system consisted of two hand actions: pinching the right hand to rotate and pinching the left hand while raising or lowering the right hand to change the maximum z stack value. After a brief ($t < 30$ s) familiarization period, subjects were given instructions to put into focus specific parts of the volumetric data, using a combination of both the rotation and max z stack features. Directly afterward, an in-person survey on the experience was collected. Due to time constraints on the day of the procedure, only a brief survey could be performed. As such, the survey structure simply involved subjects completing the System Usability Scale (SUS)[22] questionnaire, followed by verbal responses from the subjects on their general thoughts about the system and usability. The SUS was chosen as the main quantitative metric due to its brevity and the long-standing empirical evidence backing its effectiveness.[23]

The final mean of the SUS scores was 79.72 ($n = 9$, SD = 8.53). A visual distribution of the scores can be seen in Figure 5. Based on the recommended ranges developed by Bangor et al.[23] in their analysis of SUS results, a mean of 79.72 places the tested system in the "acceptable" acceptability range, which is between "good" and "excellent" in the adjective rating. A per-question breakdown of the distribution of results from each subject can be seen in Figure 6. Overall, based on the results from the SUS questionnaire, the developed system tested with the subjects had achieved an acceptable level of usability.

Subject feedback on the system design consisted of several themes. The most common positive point of feedback was the personal novelty of the hand tracking system. During the study, participants commonly played around with the representation of their virtual hands, testing how accurately the camera and model can represent their inputs. The main sticking point for most subjects was the hand input for modifying the max z stack. This was confusing to use for some and occasionally required repeated explanation. Based on this feedback, in the final version presented in this paper we have replaced the hand input for modifying the max z stack with input for changing the zoom of the camera. Users now pinch both hands and drag them in or out on the horizontal axis to bring the volume closer or further away. This change, based on the user study, was further approved as a positive development in terms usability by our non-expert users.

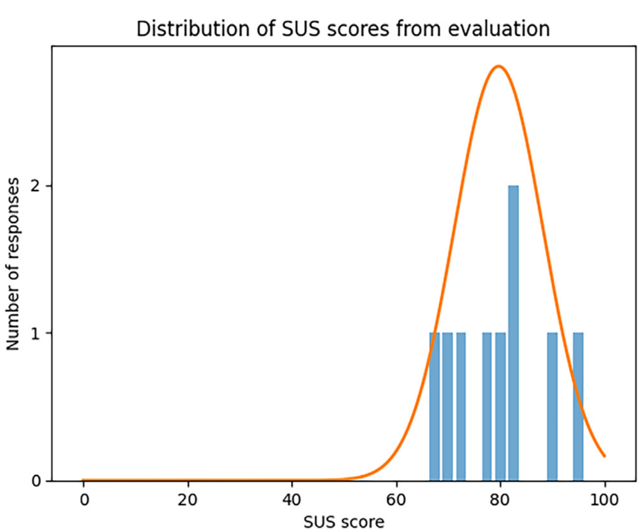


Figure 5. Distribution of final scores from SUS questionnaire ($n = 9$, mean = 79.72, SD = 8.53)

Performance

QuickVol was designed with real-time usage in mind. Baseline requirement for performance includes being able to render a new screen at or faster than 10 frames per second (FPS) on the lowest-end testing device for the included sample volume data. On the upper end, the browsers being used for testing on the devices cap the render rate to the screen refresh rate. This refresh rate is only important for the immersive or hand tracking mode, as the standard mode only renders a new frame on demand (based on any input changes).

Two devices were used in testing the high-end and low-end performance. The high-end device was a desktop computer

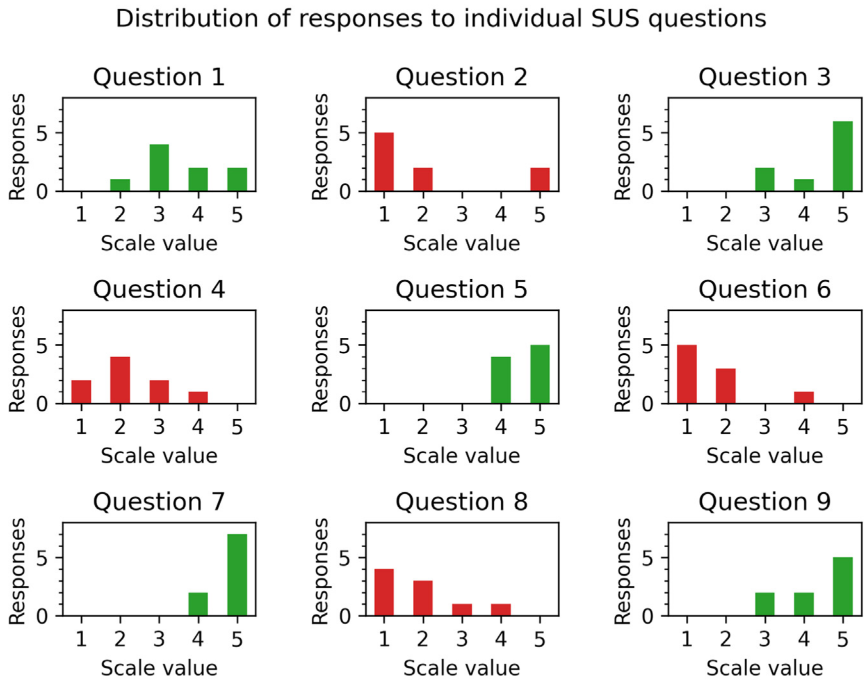


Figure 6. Per-question distribution of results from SUS questionnaire ($n = 9$)

The horizontal axis represents the individual scale value given where 1 = Strongly Disagree and 5 = Strongly Agree. Plots are grouped by color from the score calculation formulas used in the final SUS score calculation.

with an Intel Core i7-7700K @ 4.20 Ghz, NVIDIA GeForce GTX 1070, running on a monitor with a 60 Hz refresh rate. The low-end device was a Google Pixel 6 with a variable refresh rate up to 90 Hz. The performance results from testing on these devices on a sample volume that has a dimension of 256^3 (while using the MIP render mode) can be seen in Table 1. Notably, both the low-end and high-end devices were able to render the image at the screen refresh rate; however, the FPS of the low-end device dropped significantly once hand tracking was enabled. This drop is due to the high processing power required to run the hand tracking model in real time.

Another aspect of performance that had focus for QuickVol was total download size required. On first visit to the webpage, only a single gzipped Javascript file is downloaded, with a total size of 318 kB (before optionally downloading the hand tracking model). This allows for QuickVol to be significantly quicker to load than the other web alternatives, making it convenient to embed into other webpages for presentation. Although neither of Kitware's solutions are built around embedding into a webpage, FPBioimage can be deployed in such a way. In this use case, QuickVol would greatly decrease loading time in comparison, especially if there is more than one embedding on a page. A comparison between QuickVol and other web volume renderers in terms of overall file size of downloaded scripts and number of scripts fetched can be seen in Table 2.

Table 1. Frames per second on tested devices rendering a volume of dimension 256^3

	Without tracking (fps)	With tracking (fps)
Desktop	60 (capped)	60 (capped)
Pixel 6	75–85 (capped, variable)	13–15

DISCUSSION

Results from the user study and performance testing show that QuickVol is a usable system that achieves the goals first set out. As such, the current version functions closer to a proof of concept. For wider accessibility and coverage of use cases, a number of things can be considered in future development. Primarily, further work should focus on support for multiple file formats, rather than only supporting NRRD files. This would remove the requirement of converting other formats to NRRD in the preprocessing step, decreasing work time if the initial file was not already in the NRRD format. Additionally, expanding possible analysis with more advanced visual features can help further generalize the use cases of QuickVol. The current version works in the majority of cases to quickly visualize NRRD data, but it does not currently support modes for visually impressive renders and more complex ray casting algorithms.

Hand tracking in QuickVol is in a similar developmental stage. This area is still under early development and has piqued interest in those who have tested the platform, but the feature set is kept simple to have a greater focus on analyzing the viability of the approach. Pilot tests have been successful; however, the number of current possible actions is minimized. Future work could aim toward reducing the inaccuracies in the hand position predictions, which would allow for more intricate actions to be implemented. Finally, this would also introduce a greater level of immersion within the hand tracking visualization, as the virtual hands would offer a more accurate representation of the orientation of the user's real hands.

Limitations of the study

A set of limitations around the study need to be considered. Sex and gender were not recorded during the duration of the study. Therefore, those variables may play a role in the perception of the developed system that would not be detected in the current analysis. This may be prevalent in situations where there are gender disparities in specific populations that would use the tool, leading possibly to varied personal strengths and weaknesses when assessing the usability of the system. Similarly, as we conducted a pilot study to judge initial feasibility of the system, the sample size was relatively small, and the list of metrics recorded was kept minimal. Although it is compatible for an initial judgment on system usability, a study with a larger sample size

Table 2. Comparison of compiled Javascript downloaded on first visit

	Total script(s) file size	Files fetched
QuickVol	318 kB	1
VolView	4.2 MB	12
FPBioimage	6.65 MB	18
Glance	11.17 MB	236

and a wider list of recorded metrics would be the next logical step for more robust analysis.

Conclusion

QuickVol allows for rapid visualizations of volumetric data on any platform that supports a modern web browser, including mobile phones. We introduce a control scheme for analysis, which entails single-camera hand pose tracking without the need of using a WebXR compatible device. This allows for greater accessibility of the hands-only control scheme due to not requiring specialized hardware. In our testing, subjects found the hand pose control scheme intriguing and immersive. In the current state, QuickVol is usable for a number of visualization tasks and is particularly useful for users who do not have previous experience in using the more advanced volume analysis suites. Future work should focus toward advancements in smoother hand tracking algorithms and a wider range of supported formats and visual customizations.

RESOURCE AVAILABILITY

Lead contact

Further information for resources and materials should be directed to the lead contact, Maxim Kuznetsov (maakuzne@ucsc.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All experimental data required for reanalysis is listed and described within the main text.
- Original code is linked in the Code Availability section of the main text and in the STAR Methods, key resources table.
- Additional details can be provided from the lead contact upon request.
- A repository containing the source code of the project can be found at https://github.com/emkooz/QuickVol.
- Data supporting the presented results are available within the paper.

ACKNOWLEDGMENTS

We would like to thank Jesus Gonzalez-Ferrer and Julian Lehrer for providing the UMAP coordinates of their scRNA sequencing experiments. This grant was supported by Schmidt Futures (SF857) to M.T.; National Human Genome Research Institute (1RM1HG011543) to M.T.; National Science Foundation (NSF2134955 and NSF2034037) to M.T.; and the National Institute of Mental Health (1U24MH132628) to M.A.M.-R.

AUTHOR CONTRIBUTIONS

All authors envisioned the experiments. M.K. performed all experiments. M.T., M.A.M.-R., and S.K. supervised the work. M.K. wrote the manuscript with contributions from all authors.

DECLARATION OF INTERESTS

The authors declare no conflicts of interest.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
  - Ethics statement
  - Participants
- METHOD DETAILS
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - SUS questionnaire calculation
  - Lightsheet microscopy data
  - Single cell RNA sequencing data

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.111379.

Received: May 8, 2024
Revised: August 13, 2024
Accepted: November 11, 2024
Published: November 15, 2024

REFERENCES

1. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: An Open-Source Platform for Biological-Image Analysis. Nat. Methods *9*, 676–682. https://doi.org/10.1038/nmeth.2019.
2. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., et al. (2012). 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. Magn. Reson. Imaging *30*, 1323–1341. https://doi.org/10.1016/j.mri.2012.05.001.
3. Microscopy Image Analysis Software - Imaris. https://imaris.oxinst.com/.
4. REAL3D (2020). Real3d VolViCon.
5. Çöltekin, A., Lochhead, I., Madden, M., Christophe, S., Devaux, A., Pettit, C., Lock, O., Shukla, S., Herman, L., Stachoň, Z., et al. (2020). Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions. ISPRS Int. J. Geo. Inf. *9*, 439. https://doi.org/10.3390/ijgi9070439. https://www.mdpi.com/2220-9964/9/7/439.
6. Chatterjee, S. (2024). Free-Form Shape Modeling in XR: A Systematic Review. Preprint at arXiv. https://doi.org/10.48550/arXiv.2401.00924. http://arxiv.org/abs/2401.00924.
7. Rodriguez, F.C., Krapp, L., Peraro, M.D., and Abriata, L. (2023). HandMol: Coupling WebXR, AI and HCI Technologies for Immersive, Natural, Collaborative and Inclusive Molecular Modeling. https://chemrxiv.org/engage/chemrxiv/article-details/6561ab3829a13c4d47f1eaf4.
8. Xu, J., Thevenon, G., Chabat, T., McCormick, M., Li, F., Birdsong, T., Martin, K., Lee, Y., and Aylward, S. (2023). Interactive, in-Browser Cinematic Volume Rendering of Medical Images. Comput. Methods Biomech. Biomed. Eng. Imaging Vis. *11*, 1019–1026, Visited on 06/05/2023). https://doi.org/10.1080/21681163.2022.2145239.
9. Kitware Glance. https://github.com/Kitware/glance.
10. Kitware/Vtk-Js. Kitware, Inc. (2024). https://github.com/Kitware/vtk-js.
11. Kitware/ParaView. Kitware, Inc. (2024). https://github.com/Kitware/ParaView.

12. Fantham, M., and Kaminski, C.F. (2017). A New Online Tool for Visualization of Volumetric Data. Nat. Photonics *11*, 69. https://doi.org/10.1038/nphoton.2016.273.
13. Bienroth, D., Charitakis, N., Wong, D., Jaeger-Honz, S., Garkov, D., Watt, K.W., Stolper, J., Chambers-Smith, H., MacGregor, D., Christiansen, B., et al. (2023). Automated Integration of Multi-Slice Spatial Transcriptomics Data in 2D and 3D. Preprint at bioRxiv. https://doi.org/10.1101/2023.03.31.535025.
14. Legetth, O., Rodhe, J., Lang, S., Dhapola, P., Wallergård, M., and Soneji, S. (2021). CellexalVR: A virtual reality platform to visualize and analyze single-cell omics data. iScience *24*, 103251. https://doi.org/10.1016/j.isci.2021.103251.
15. Stein, D.F., Chen, H., Vinyard, M.E., Qin, Q., Combs, R.D., Zhang, Q., and Pinello, L. (2021). singlecellVR: interactive visualization of single-cell data in virtual reality. Front. Genet. *12*, 764170. https://doi.org/10.3389/fgene.2021.764170.
16. Three.Js – JavaScript 3D Library. https://threejs.org/.
17. Blinn, J.F. (1977). Models of Light Reflection for Computer Synthesized Pictures. SIGGRAPH Comput. Graph. *11*, 192–198. https://doi.org/10.1145/965141.563893.
18. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., and Grundmann, M. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. Preprint at arXiv. https://doi.org/10.48550/arXiv.2006.10214.
19. WebXR Hand Input Module - Level 1. https://www.w3.org/TR/webxr-hand-input-1/.
20. MozillaReality/WebXR-emulator-extension at HandWIP. https://github.com/MozillaReality/WebXR-emulator-extension/tree/HandWIP.
21. Teem: Nrrd. https://teem.sourceforge.net/nrrd/.
22. Brooke, J. (1996). SUS: A 'Quick and Dirty' Usability Scale. In Usability Evaluation In Industry (CRC Press).
23. Bangor, A., Kortum, P.T., and Miller, J.T. (2008). An Empirical Evaluation of the System Usability Scale. Intern. J. Human Comput. Interact. *24*, 574–594. https://doi.org/10.1080/10447310802205776.
24. Mostajo-Radji, M.A., Mancia Leon, W.R., Breevoort, A., Gonzalez-Ferrer, J., Schweiger, H.E., Lehrer, J., Zhou, L., Schmitz, M.T., Perez, Y., Mukhtar, T., et al. (2024). Fate Plasticity of Interneuron Specification. Preprint at bioRxiv. https://doi.org/10.1101/2024.10.02.614266. https://www.biorxiv.org/content/early/2024/10/03/2024.10.02.614266.full.pdf.
25. Susaki, E.A., Tainaka, K., Perrin, D., Yukinaga, H., Kuno, A., and Ueda, H.R. (2015). Advanced CUBIC protocols for whole-brain and whole-body clearing and imaging. Nat. Protoc. *10*, 1709–1727. https://doi.org/10.1038/nprot.2015.085.
26. Velasco, S., Kedaigle, A.J., Simmons, S.K., Nash, A., Rocha, M., Quadrato, G., Paulsen, B., Nguyen, L., Adiconis, X., Regev, A., et al. (2019). Individual brain organoids reproducibly form cell diversity of the human cerebral cortex. Nature *570*, 523–527. https://doi.org/10.1038/s41586-019-1289-x.
27. Baudin, P.V., Sacksteder, R.E., Worthington, A.K., Voitiuk, K., Ly, V.T., Hoffman, R.N., Elliott, M.A.T., Parks, D.F., Ward, R., Torres-Montoya, S., et al. (2022). Cloud-controlled microscopy enables remote project-based biology education in underserved Latinx communities. Heliyon *8*, e11596. https://doi.org/10.1016/j.heliyon.2022.e11596.
28. Pollen, A.A., Bhaduri, A., Andrews, M.G., Nowakowski, T.J., Meyerson, O.S., Mostajo-Radji, M.A., Di Lullo, E., Alvarado, B., Bedolli, M., Dougherty, M.L., et al. (2019). Establishing cerebral organoids as models of human-specific brain evolution. Cell *176*, 743–756.e17. https://doi.org/10.1016/j.cell.2019.01.017.
29. Gonzalez-Ferrer, J., Lehrer, J., O'Farrell, A., Paten, B., Teodorescu, M., Haussler, D., Jonsson, V.D., and Mostajo-Radji, M.A. (2024). SIMS: A deep-learning label transfer tool for single-cell RNA sequencing analysis. Cell Genom. *4*, 100581. https://doi.org/10.1016/j.xgen.2024.100581.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
N,N,N′,N′-Tetrakis(2-hydroxypropyl)ethylenediamine	Tokyo Chemical Industry	Cat# T0781
Syto16	Thermo Fisher Scientific	Cat#: S7578
Urea	Millipore Sigma	Cat#: U5378
Paraformaldehyde	Thermo Fischer Scientific	Cat#: 043368.9
Triton X-100	Millipore Sigma	Cat#: X100
Critical commercial assays		
Chromium Single Cell 3′ Library and Gel Bead Kit v.2	10× Genomics	PN-120237
Deposited data		
Lighsheet images of brain organoids	Mostajo-Radji et al.[24]	N/A
Single cell RNA sequencing data of brain organoids	Velasco et al.[25]	https://singlecell.broadinstitute.org/single_cell/study/SCP282/reproducible-brain-organoids#study-summary
Software and algorithms		
Code of developed system	Authors	https://github.com/emkooz/quickvol
Other		
UMAP coordinates of single cell RNA sequencing data	Gonzalez-Ferrer et al.[26]	N/A

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

Ethics statement

The work proposed here was approved by the University of California Santa Cruz Institutional Review Board. Informed consent was obtained from all participants.

Participants

All the participants were students at Alisal High School in Salinas, California. The students have previously worked with our group performing remote project-based education using cloud technologies, as described in Baudin et al., 2022[27] and therefore have been familiarized in computational analysis of biological data. In this high school, over 97% of students self-identify as Hispanic. We did not record any gender information of the participants.

METHOD DETAILS

Data collection for the user study described in the main text was completed on a Dell Inspiron G7 7500. The relevant specifications of this device are the Intel Core i9-10886H CPU @ 2.40 GHz and the NVIDIA GeForce RTX 2070 GPU. Compared to the other devices used for data collection and analysis, this device positions itself closer to the desktop computer in capability. All other relevant details are described in the main text.

QUANTIFICATION AND STATISTICAL ANALYSIS

SUS questionnaire calculation

To evaluate QuickVol in our pilot study, the System Usability Scale (SUS)[22] was employed. The SUS questionnaire was used as it allowed rapid collection of data on the usability of the system within a small time period. Greater background and discussion on the decision to use the SUS can be found in the main text under the results section. The final SUS score was calculated as follows:

$$SUS = \frac{\sum_{k=0}^{subjects} 2.5\left(\sum_{n=1}^{4} 5 - \mathrm{score}_k(2n) + \sum_{n=0}^{4} \mathrm{score}_k(2n+1) - 1\right)}{subjects}$$

where *subjects* is the total number of responses to the questionnaire and $score_k(n)$ is the individual score given for question n. The final calculations and the individual responses used in the calculations are described in the main text.

Lightsheet microscopy data

We tested the use of QuickVol in biological experiments by reconstructing lighsheet images of brain organoids. Brain organoids for visualization were generated as described in Pollen et al., 2019.[28] The images used have been described in Mostajo-Radji et al., 2024.[24] A subset of the cells were labeled with the genetic indicator td-Tomato. The organoids were fixed at room temperature for 45 min in 4% paraformaldehyde (Thermo Fischer Scientific, 043368.9). For nuclear staining we used Syto16 green fluorescent dye (Thermo Fisher Scientific, S7578).

Whole organoid clearing was performed using ScaleCUBIC-1 solution as described in Susaki et al., 2015.[25] Briefly, the solution contained: 25% wt Urea (Millipore Sigma, U5378), 25% wt N,N,N′,N′-Tetrakis(2-hydroxypropyl) ethylenediamine (Tokyo Chemical Industry, T0781) and 15% Triton X-100 (Millipore Sigma, ×100) dissolved in distilled water. Organoids were incubated in ScaleCUBIC-1 solution overnight at room temperature at 90 rpm. Whole organoid imaging was performed using a custom made Lattice Light Sheet Microscope (UCSF Biological Imaging Development Center) and the images were deconvoluted using Richardson-Lucy algorithm. An example of loading the brain organoid data in QuickVol can be seen in Figure S1.

Single cell RNA sequencing data

As another real implementation of the visualization, we tested the applicability of QuickVol to reconstruct UMAP plots of single cell RNA (scRNA) data. The dataset used was derived from 6-months old brain organoids described in Velasco et al.[26] The dataset was generated using Chromium Single Cell 3′ Library and Gel Bead Kit v.2 (10× Genomics, PN-120237) and sequenced using the Illumina NextSeq 500 instrument. We downloaded the dataset from the Single Cell Portal administered by the Broad Institute. https://singlecell.broadinstitute.org/single_cell/study/SCP282/reproducible-brain-organoids#study-summary. For visualization, we used the same UMAP coordinates described in Gonzalez-Ferrer et al., 2024.[29] An example of loading the single cell RNA sequence data can be seen in Figure S2.