Multiuser virtual reality environment for visualising neuroimaging data

David W. Shattuck ✉

Ahmanson-Lovelace Brain Mapping Center, Department of Neurology, David Geffen School of Medicine at UCLA, Los Angeles, California, USA
✉ *E-mail: dshattuck@ucla.edu*

Published in Healthcare Technology Letters; Received on 13th August 2018; Accepted on 20th August 2018

The recent advent of high-performance consumer virtual reality (VR) systems has opened new possibilities for immersive visualisation of numerous types of data. Medical imaging has long made use of advanced visualisation techniques, and VR offers exciting new opportunities for data exploration. The author presents a new framework for interacting with neuroimaging data, including MRI volumes, neuroanatomical surface models, diffusion tensors, and streamline tractography, as well as text-based annotations. The system was developed for the HTC Vive using C++, OpenGL, and the OpenVR software development kit. The author developed custom GLSL shaders for each type of data to provide high-performance real-time rendering suitable for use in a VR environment. These are integrated with an interface that enables the user to manipulate the scene through the Vive controllers and perform operations such as volume slicing, fibre track selection, and structural queries. The software can read data generated by existing automated brain MRI analysis packages, enabling the rapid development of subject-specific visualisations of multimodal data or annotated atlases. The system can also support multiple simultaneous users, placing them in the same virtual space to interact with each other while visualising the same datasets, opening new possibilities for teaching and for collaborative exploration of neuroimaging data.

1. Introduction: In the past few years, a number of high-performance consumer virtual reality (VR) systems have been released, stimulating great interest in the development of immersive visualisation for numerous types of data. Medical imaging has a long history of using advanced visualisation techniques and is a natural application for these new tools. Examples of early work in medical imaging VR for data analysis include the exploration of VR in the interpretation of diffusion tensor imaging (DTI) data [1]. Such early efforts to develop VR for biomedical visualisation typically required highly specialised hardware, such as CAVE-style projection systems [2]. More recently, the release of widely-available, commercial head-mounted systems for VR, such as the Oculus Rift and the HTC Vive, has resulted in the development of several neuroimaging VR packages that make use of these systems. Some recent examples include Precision VR (Surgical Theater, LLC, Mayfield Village, OH, USA), a commercial product targeted at surgical planning; Neuro Imaging in VR (NIVR) [3], a Unity-based system that makes use of pre-processed and pre-rendered data to produce a visual experience to explore MRI; and the Virtual Brain Segmenter (VBS) [4], a Unity-based system for making corrections to brain segmentation data in VR. The HTC Vive has been integrated with the existing medical image processing framework MeVisLab, also making use of the OpenVR software development kit (SDK) [5]. Kitware, Inc. (Clifton Park, NY, USA) has also added support for OpenVR to the Visualisation Toolkit (VTK) [6] and recently developed an extension for 3D Slicer that uses these OpenVR modules to interact with a scene in that software package [7].

In this Letter, we present an HTC Vive-based VR framework that we have developed for interacting with brain imaging data, including 3D image volumes, surface models, diffusion tensors, and streamline tractography. Our VR system is also capable of supporting multiple simultaneous users, enabling a shared experience and opening new possibilities for data exploration and teaching with brain imaging and other biomedical imaging data. The system was developed using open libraries and can support a variety of file types. We created this framework with the primary goal of providing a VR environment for exploring data processed by our BrainSuite software package (http://brainsuite.org) [8], which we have been developing and supporting for the past two decades. BrainSuite is a collection of tools for analysing and visualising structural and diffusion MRI data, with the ability to perform various tasks including automatically extracting cortical surface models [8], registering and labelling surface and volume data based on a labelled reference brain MRI [9], and processing diffusion MRI data [10, 11]. The BrainSuite graphical user interface provides an array of visualisation capabilities that enable users to interact with volume and surface data, display diffusion tensors or orientation distribution functions (ODFs), delineate regions of interest (ROIs) in an image volume, or delineate sulci on surface models of the brain. These tools have been used for a variety of applications, including group analysis studies (e.g. [12, 13]) and constructing brain atlases (e.g. [14, 15]).

For this work, we focused our efforts on bringing BrainSuite's core visualisation capabilities into the VR framework with the goal of providing utility for a few key applications. The most direct use of this work is the visualisation of the results of image processing workflows, including the segmentation of anatomical structures in MRI data or extracted surfaces or the analysis of statistical results in group studies. These new VR tools also offer an opportunity to gain a better understanding of how algorithms perform, which may help us to improve components of our MRI processing software. Another application where we anticipate major benefits from VR is the analysis of tractography data. Software packages such as TrackVis [16], 3D Slicer [17], and BrainSuite provide tools for extracting bundles of streamlines from whole-brain tractography though the placements of ROIs. These bundles can then be used for subsequent analyses of white matter pathways or to visually understand the connectivity of the brain. Placing these ROIs using a traditional 3D desktop display can be difficult, as can interpreting the paths of streamlines, even when provided in an anatomical context. VR provides new opportunities for interacting with tractography data in virtual space, which may be more intuitive to users; we developed new capabilities for such interactivity in our framework (see Section 2.2.6). A third application we have for this framework is to produce interactive brain atlases, in which users can explore an annotated dataset with the ability to perform spatial queries that return detailed information about structures at a given location. These capabilities provide new opportunities for teaching, particularly when paired with the ability to support multiple users in the same virtual space. In a general sense, the goal of our work is to provide mechanisms for improved understanding of the 3D relationships between neuroanatomical structures and related data, and to provide these capabilities in an environment in which multiple users, operating either in the same physical space or remotely, can interact with the same data as well as each other. We describe here our technical progress towards building such a framework.

2. Methods: We developed our neuroimaging VR system for the HTC Vive hardware using Microsoft Visual Studio 2015, C++, OpenGL, and the OpenVR SDK (https://github.com/ValveSoftware/openvr) developed by Valve Corporation (Bellevue, WA, USA). Though game development platforms such as Unity are often used to build new VR platforms, we opted to use OpenVR because it provides a C++ API that was easy to integrate with our existing codebase. The SDK also includes example code for a basic VR application, which we used as a starting point for our own software. In addition, the OpenVR SDK is licensed under a BSD license, enabling its free use in our platform. Though we have focused our development to date on the HTC Vive headset, the OpenVR API generalises the interface for interacting with a variety of VR displays and controllers, which enables developers to target multiple brands of headsets without having to write specialised code for each device. In principle, this will enable our software to be readily ported to other popular VR platforms, such as the Oculus Rift, without major development efforts. Our system also makes use of the Simple DirectMedia Layer (SDL) library (https://www.libsdl.org/), as well as a related package for rendering True Type fonts (SDL_ttf).

We based our design largely on our past experience creating BrainSuite, and we made use of C++ libraries that we developed previously for it. This enabled us to readily use our existing readers for widely used neuroimaging file formats, such as NIfTI, as well as some of our own proprietary formats for specialised types of data. This also provided us with functionality for performing various data operations, including filtering streamlines with sets of ROIs. In principle, we could have used VTK as the basis for our VR framework, as it has many powerful features for rendering the types of data that we support in our visualisation system, as well as support for OpenVR. However, this likely would have required considerable effort to integrate with our existing BrainSuite libraries and data structures. Using our own libraries, in combination with resources provided by the OpenVR SDK, enabled us to rapidly develop an initial VR application for visualising cortical surface models generated by BrainSuite, which we then expanded to include a wider range of data types. In addition, writing customised GLSL shaders allows us to perform very efficient GPU-based operations for changing aspects of the data display, such as highlighting ROIs in a surface model (see Section 2.2.3).

2.1. Rendering engine: We focused our development efforts on creating new rendering components for the primary data types that we typically visualise in BrainSuite. These included surface models represented as triangle meshes; streamline tractography data, represented by sets of vectors of points; image volumes, represented as 3D matrices of scalar or RGB voxels; and diffusion tensor volumes, represented sets of eigenvectors and eigenvalues at each voxel location. While our desktop-based BrainSuite software supports these types of rendering, it was developed for a broader range of platforms supporting older versions of OpenGL. To achieve high performance on our new VR system, we produced a new set of code using vertex and fragment shaders developed using GLSL 4.10.

2.1.1 Surface models: Surface models are used routinely in neuroimaging to represent the boundaries of the cerebral cortex or other neuroanatomical structures, such as the basal ganglia or hippocampus. A number of packages exist that can automatically process a 3D T1-weighted MRI to generate cortical surface models and label them according to an anatomical structure (e.g. [9, 18, 19]). The typical data structure for these surfaces consists of vertices and triangle indices, which can be readily stored on the GPU for fast rendering. We also may have additional data associated with each vertex, including an integer value denoting the anatomical ID of the structure to which the vertex belongs and an RGB colour associated with that vertex, which can indicate anatomical structure, cortical thickness measurements, or statistical parameters. Vertex colour and anatomical identifier data are also stored on the GPU. The default representation of the surface is then a coloured mesh (see Fig. 1), but we can also use the anatomical ID to isolate ROIs, as shown in Fig. 2. We implemented three

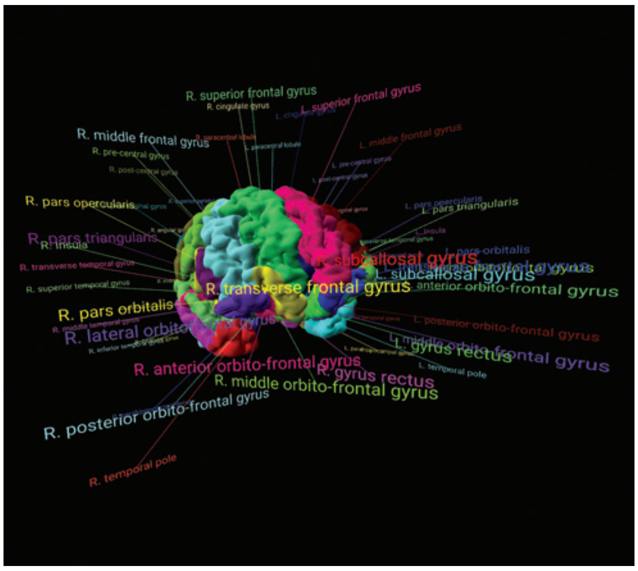
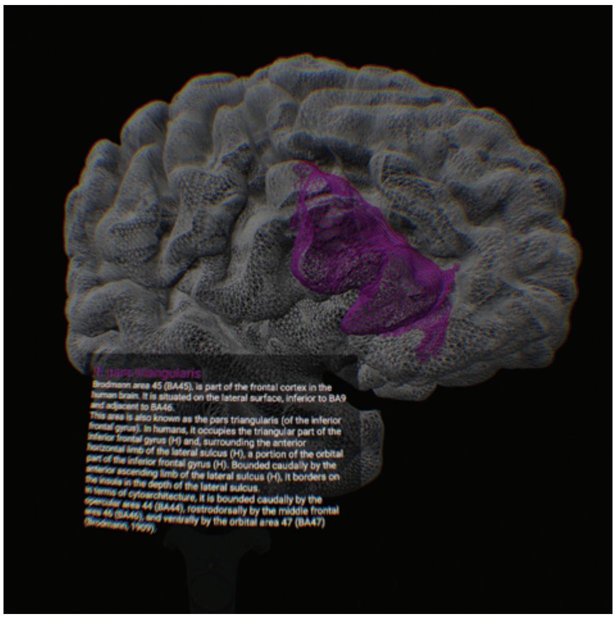


Fig. 1 *ROI labelling. Vive screenshot of a mid-cortical surface model automatically extracted and labelled using a reference atlas. ROI labels are automatically positioned and reorient towards the viewer as the viewer's head position changes*


R. pars triangularis

Brodmann area 45 (BA45), is part of the frontal cortex in the human brain. It is situated on the lateral surface, inferior to BA9 and adjacent to BA46.
This area is also known as the pars triangularis (of the inferior frontal gyrus). In humans, it occupies the triangular part of the inferior frontal gyrus (H) and, surrounding the anterior horizontal limb of the lateral sulcus (H), a portion of the orbital part of the inferior frontal gyrus (H). Bounded caudally by the anterior ascending limb of the lateral sulcus (H), it borders on the insula in the depth of the lateral sulcus.
In terms of cytoarchitecture, it is bounded caudally by the opercular area 44 (BA44), rostrodorsally by the middle frontal area 46 (BA46), and ventrally by the orbital area 47 (BA47) (Brodmann, 1909).

Fig. 2 *Structure query. In Structure Query mode, the user can position the controller within a structure and pull its trigger to select a structural ROI. The software highlights that region by shading the other structures in grey. Labelled ROIs can also be associated with descriptive text, which is shown to the user in the form of a virtual tablet. Shown is the pars triangular is, with descriptive text extracted from Wikipedia. The image appears distorted due to corrections applied for display in the Vive system*

different modes of surface display: solid, wireframe mesh (edges only), and point mode (vertices only).

2.1.2 Structural labels: We implemented a system for attaching text labels to structures in the brain (see Fig. 1). Each label is rendered to a texture using the SDL_ttf library and displayed with a line that connects the text to a point in the image space. Given a labelled brain surface, these labels can be calculated automatically based on the centroid of the vertices that share that label, or they can be specified manually. The labels are positioned at a fixed radius from the centre of the brain and rotate automatically so that they face the viewer's display.

2.1.3 Slice-based volume rendering: We implemented slice-based volume rendering to display volumetric imaging data [20]. To achieve high performance, volumetric data are stored on the GPU in a 3D texture buffer. A series of planes facing the user are then intersected with the volume, textured with the data, and rendered with translucency to achieve a volumetric display (see Fig. 3). Our system can render greyscale and colour RGB NIfTI volumes.

2.1.4 Diffusion tensor display: Diffusion magnetic resonance imaging measures the diffusion properties of water in the brain or other tissues and is widely used to characterise the structure of white matter [22]. DTI captures the properties of water diffusion at each voxel using a rank-2 tensor, which provides information about the rate of diffusion in each direction and can be used to infer the fibre tract orientation [23]. Diffusion tensors are frequently represented visually using ellipsoidal glyphs, with the axes determined by the eigenvectors and eigenvalues of the tensor (see Fig. 4). The glyphs may be coloured according to the direction of the major eigenvector, with (x, y, z) components converted to (r, g, b) values after taking their absolute values. A related option is to colour the glyph based on the fractional anisotropy (FA) measure, or a combination of direction and FA. We implemented ellipsoidal glyphs on our system using a shader-based approach. A single vertex array object is used to store a tessellated sphere on the GPU. At each voxel being displayed, we deform this instance based on the eigenvectors and eigenvalues. In our present implementation, we transmit the eigenvectors and eigenvalues for each voxel being displayed every frame and perform the tensor deformation in the shader. Though this requires 12 32-bit floating-point values per voxel, we are still able to achieve acceptable frame rates when rendering full tensor slices (see Fig. 4). In addition, we store multiple instances of the sphere at different levels of detail, which can be selected during run-time. The colour of the glyph is computed in the shader and can be set to: (i) the first eigenvector after taking an element-wise absolute value; (ii) the FA, computed by the shader from the eigenvalues; or (iii) the product of FA and direction (colour FA).

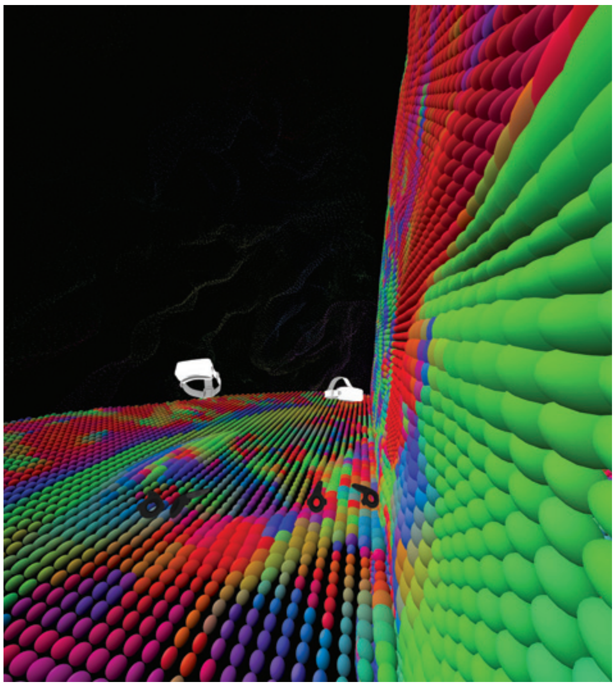


Fig. 4 *Diffusion tensor display. Two users, observed by a third, interact with a large scale diffusion tensor display*

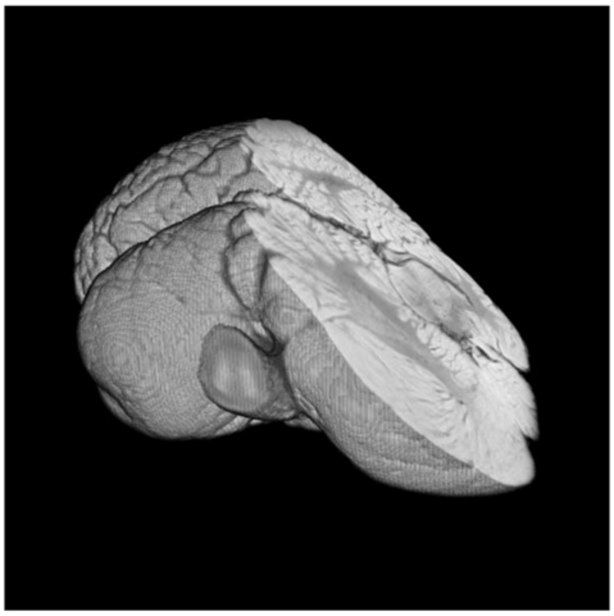


Fig. 3 *Volume rendering of a bottlenose Dolphin brain. Shown is a slice-based volume rendering of a bottlenose dolphin brain, retrieved from the brain catalogue [21]. The user can slice the volume at any angle by depressing the trigger on the controller and reorienting the controller to select a cutting plane, which updates in real time*

2.1.5 Streamline tractography: Diffusion tractography is a process that is frequently applied to diffusion tensors or ODFs to generate models of the white matter connectivity within the brain. A variety of diffusion tractography approaches exist, including deterministic (e.g. [24]) and probabilistic (e.g. [25]) methods. In our current software, we developed support for deterministic streamline tractography, where tracks are represented by sets of points that define contiguous line segments. Such tracks can be generated by a number of freely available software packages, including BrainSuite, TrackVis/Diffusion Toolkit [16], DSIStudio [26], and Dipy [27]. We assign a colour to each vertex, based on the direction of the adjacent line segments. The vertices and colours of all tracks in a tractography dataset are stored in a single vertex array object, and we can either render it in its entirety or only render subsets of tracks using OpenGL array drawing operations. An example of a tractography subset rendering is shown in Fig. 5.

2.2. User interface: We developed a set of user interface components for interacting with the types of neuroimaging data described in the previous section. A collection of imaging data, consisting of a combination of surfaces, volumes, streamline tracks, tensor volumes, and text descriptions, can be specified in a plain text file and loaded into our software system. Scenes, which comprise subsets of the data and the visualisation and interface settings used to view them, can be specified to provide presets displays. Lists of scenes can also be specified, which then allows the user to cycle through different aspects of the data, e.g.

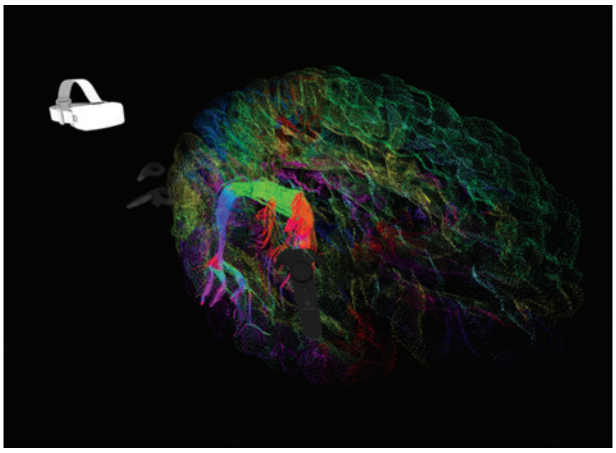


Fig. 5 *Track selection mode. A user has selected spherical several ROIs to isolate a bundle of streamlines. Also visible is a second user in the system*

viewing different systems of the brain or different types of data for the same subject.

2.2.1 Display modes: We defined four preset display modes that present the data at different scales. The first viewing mode presents an intermediately sized brain, which is scaled to occupy approximately half of the space available in each dimension in the 'play area', i.e. the usable space defined for the Vive setup. The second mode scales to fill this space, while the third mode presents the data such that it surrounds the user at a scale of approximately 300:1. In each of these modes, users can walk around or inside the brain to view different aspects of the data. The fourth view presents the data in hand-held mode, at near real-world scale and attached to the position of the hand controller. In this mode, the user can easily reposition the brain by moving the controller, with the view updating in real time.

2.2.2 Controllers: The HTC Vive uses two hand-held controllers, and each controller provides several inputs that are available from within the OpenVR SDK. These include a large circular touch pad, a small button at the top of the controller, a button on the side of the controller, and a trigger on the underside of the controller. While these provide a number of options for interaction, they are limited compared to the combination of a keyboard and a mouse. We thus bind context-specific actions to these inputs based on the available data and the current viewing state. Each controller presents the user with different actions, which are shown visually on the controller through the use of a text overlay.

The top button on the first controller will cycle through the four different display modes described above, while the top button on the second controller will cycle through different preset data scenes. Pressing the right and left sides of either touch pad will rotate the displayed data around its vertical axis. Pressing up or down on the touch pad of the first controller will enlarge or shrink the data, while these controls on the second controller will translate the data up or down vertically. The side button cycles through different possible behaviours for the trigger, which can be used to perform a structure query, apply a cutting plane, select tensor slices, and select streamline tracks. We describe each of these actions in detail below.

2.2.3 Structure Query mode: If a label index volume has been loaded into the system, it can be used to identify neuroanatomical structures at any point in the volumetric image space. In Structure Query mode, the user can select a region by depressing the trigger on the Vive controller. The system reads the structure label at that coordinate if available, presents the user with the name of the structure, and isolates the structure in the displayed surface (see Fig. 2). The system can also be encoded with additional reference text describing the structure, which is presented to the user in the form of a virtual tablet view (also shown in Fig. 2).

2.2.4 Cutting Plane mode: In Cutting Plane mode, the system will apply a cutting plane to the volume data (see Fig. 3). The cutting plane is defined by the position and orientation of the controller, enabling the user to reslice the data in real time by moving or rotating the controller while depressing the trigger. The cutting plane can also be configured to apply to the surface models.

2.2.5 Tensor Slice mode: Tensors are displayed using three orthogonal cardinal planes of data defined by a single coordinate. The user can reposition this coordinate by pressing the trigger and moving the controller; the tensor display tracks this and updates in real time.

2.2.6 Track Selection mode: Users can select subsets of tracks within the virtual environment (see Fig. 5). To do this, the user moves the controller through the virtual space while holding the trigger. A spherical ROI is displayed while the trigger is depressed and used to filter the tractography set; only tracks intersecting this spherical ROI are displayed. Filtering is performed on the CPU side, and the system updates the displayed streamlines in real time as the user moves the controller. When the user releases the trigger, this pending ROI is stored and the user can place additional ROIs to isolate particular groups of tracks.

2.3. Multi-user interaction: A major feature of our new VR framework is that it can support multiple simultaneous users operating in the same virtual space. We implemented this using a client–server model, where one user operates as the server and has control over the system display. The VR environment for each user is driven by an individual networked computer connected to the user's headset. The client user can either be in the same physical location as the server, using the same Vive setup or in a separate physical location using an individual Vive setup. Each client computer has access to a copy of the data to be displayed, which can be either on a local drive or a shared network drive. The server listens for clients on a TCP/IP port and then establishes a network connection with each client. Once a client is connected, the server will routinely transmit a small data object that contains viewing state information, which each client uses to update its display. This object includes data necessary to synchronise the view, such as rotation, scale, volume position, and cutting plane. These objects are transmitted every frame on high-speed network connections, but this rate can be reduced for slower connections. Each client transmits pose information for its headsets and controllers to the server, which broadcasts these to the other clients for display. Models of the headsets and controllers for the other users in the system are rendered in each individual client view (see Figs. 4 and 5). This enables users to interact directly with each other and also helps to avoid real-world collisions between users operating in a shared physical space.

Additional data are transmitted when specific actions are taken by the server. During Structure Query mode, the structure query result is shown attached to each client's individual controller, enabling each user to read the result individually. Similarly, when the server selects the hand-held mode, the imaging data are rendered attached to the client's controller so that each user can manually reorient the data. During Track Selection, the server will also transmit the positions and sizes of the spherical filter ROIs, which each client then uses to filter the tractography. This enables all clients to view the track selection process in real-time as the server selects subsets of tracks (see Fig. 5). When the server cycles through different data modes, these states are also transmitted to the clients so that their views are synchronised with the server.

3. Results: We implemented our VR brain imaging system as described above and created datasets for use with it. We conducted a performance evaluation using a desktop computer (Intel i7-8700k, 64 GB RAM, NVidia GeForce GTX 1080Ti graphics card) and two datasets that included structural T1-weighted MRI and diffusion MRI data. The first was retrieved from the Beijing Normal University, State Key Laboratory of Cognitive Neuroscience and Learning Enhanced Sample, part of the 1000 Functional Connectomes Project (http://fcon_1000.projects.nitrc.org/indi/retro/BeijingEnhanced.html). The second was a higher resolution dataset retrieved from the Human Connectome Project [28]. We processed each dataset using BrainSuite's automated neuroimage analysis tools to generate a set of coregistered T1 and DTI volumes, a labelled anatomical index volume, cortical and subcortical surface models, and deterministic streamline tractography. We computed frame rates for each datatype by setting a fixed viewpoint, rotating the displayed data by one degree about the vertical axis during each frame, measuring the time required to display 3600 frames, and dividing the number of frames by that result. While using a fixed viewpoint enables us to perform a consistent evaluation, we do note that rendering performance is dependent on the viewpoint. A more thorough evaluation might consider multiple positions; however, this study does provide some insights into system limitations. Surface benchmarks were computed using cerebral cortical surface models. Volume rendering was performed using a colour FA image for the Beijing data, and a greyscale MRI image for the HCP data; up to 512 slices used for the rendering, depending on the viewing angle. Tensor glyphs were rendered for each voxel labelled as the brain for three orthogonal slices centred in the image volume. Fibre track benchmarks were computed using whole-brain tractography. Data sizes and average frame rates are detailed in Table 1. Our software achieved suitable frame rates for surface models for both datasets, while the frame rate for volume rendering decreased to approximately two-thirds the desired 90 frames per second. Glyph rendering for tensors was one of the slower operations, particularly for the HCP data. One possible improvement for this would be to store the tensor eigenvalues and eigenvectors in GPU memory rather than transmitting them each frame. Streamline data for both datasets rendered at suitable frame rates, but we note that under other conditions, we have at times observed lag during track filtering operations and with denser tractography datasets. This could potentially be addressed by downsampling the streamline data.

Though we have used our system on a variety of data, including non-brain data, we demonstrate its functionality on the single-subject multi-modality dataset from the Beijing Normal University. In addition to the T1-MRI and DTI volumes, labelled anatomical volume, and labelled cortical surface models, we also produced a set of anatomical descriptions for the atlas labels, which were extracted from the Wikipedia entries for the labelled structural ROIs. Figs. 1, 2, 4 and 5 show example views from this specific dataset. We note that though most of these images each emphasise only a single mode of data, any of the data can be presented simultaneously in our framework. For example, Fig. 5 shows a subset of tracks displayed inside a cortical surface model that was rendered in point mode, which can provide an anatomical reference during track selection. The use of this collection of data in our VR system represents a multi-modal interactive brain atlas that users can explore, reslice, dissect, and query to learn more about neuroanatomy. Importantly, this dataset was created with freely-available automated software that could be readily applied to other MRI data to produce new, subject-specific atlases that can be brought into our software to explore their individual neuroanatomical details.

We have thus far used the software in two different classroom settings to present MRI-based neuroanatomy to undergraduate and graduate students at UCLA. In both of these cases, we made use of the multiuser mode and were able to support four simultaneous users with the software. Though we have not yet performed a formal user evaluation of the software, we describe some of our qualitative observations here. Even though the users saw only the renderings of the headsets and controllers of the others in the system, the real-time tracking of the user positions produced natural human movements, which enhanced the ability of the users to interact with each other. In other uses of the software, users experienced with diffusion tractography reported that it was much easier to understand the paths the tracks followed and relate them to anatomical structures using our software as compared to conventional desktop frameworks.

Table 1 Rendering performance

	Beijing dataset	HCP dataset
voxel dimensions	$128 \times 256 \times 256$	$260 \times 311 \times 260$
voxel resolution, mm^3	$1.33 \times 1.0 \times 1.0$	$0.7 \times 0.7 \times 0.7$
number of surface triangles	425,848	1,080,156
number of tensor glyphs	35,371	82,011
number of fibre tracks	31,770	49,862
number of fibre track line segments	8,158,976	14,882,770
surface rendering (frames/s)	89.524	89.524
volume rendering (frames/s)	60.095	60.763
glyph rendering (frames/s)	29.841	15.587
fibre track rendering (frames/s)	89.524	89.524

4. Discussion: In this work, we have presented a new multi-user framework for visualising brain imaging data in VR. The framework supports data from structural and diffusion MRI, providing visualisation of surface models, image volumes, diffusion tensors, and streamline tractography. A key aspect of our system is that it can support multiple simultaneous users in a virtual space, which opens new opportunities for education and collaboration. A second novel feature of our software is the ability to perform an interactive ROI-based selection of streamline data, which provides a new mechanism for performing virtual dissection of diffusion tractography. Though still in a relatively early stage of development, our VR framework already provides the tools necessary to create explorable atlases with annotated anatomical references. The system is highly flexible and can import data that have been preprocessed with freely available neuroimaging tools, enabling the rapid development of customised visualisation experiences of new data. Importantly, these data need not be specific to neuroimaging, as our system can also support visualisation of data from other anatomical structures, such as cardiac MRI. We are also exploring potential clinical applications, including surgical planning for deep brain stimulation.

An important aspect of this work that remains to be done is a set of formal evaluations of the framework's utility. One of the most promising applications of our VR software is its use in isolating bundles of diffusion streamlines representing specific tracts. We plan to conduct an evaluation to compare how human raters perform, in terms of speed, accuracy, and inter-rater reliability, when completing this task. Specifically, we will compare the use of our system versus existing interactive desktop programs that provide this functionality, including BrainSuite and TrackVis. Second, we are interested in evaluating our VR platform as a tool for education. We anticipate that our system will lead to improved understanding of the spatial relationships of neuroanatomical structures, as well as the connectivity between them, which we expect will enhance trainees' abilities to identify these in neuroimaging data. We also anticipate that providing reference information in the VR system will enhance the learning experience, with particular application to learning functional neuroanatomy. We plan to evaluate students to compare training in these conditions versus more traditional methods, which may include textbooks, laboratory work, computer-based training, or lectures. Third, we plan to conduct user studies to evaluate the usability of the system, in particular, to evaluate how we may make the system configurable for individual users, including improvements in accessibility. An important aspect of these evaluations will be to determine perceptual limitations in using our system and identify potential solutions for them. Some of the aspects that we intend to study include how well users can judge distances and scale when interpreting spatial relationships within the brain, what frame rates are required for users to have a useful experience in our system, whether users experience motion sickness while using the system, and what types of interaction and feedback provide users with the best experience.

There are several additional directions that we plan to explore as we work towards a public release of the system. We are currently working with instructors at UCLA to develop VR modules for neuroanatomy courses to provide students with new learning experiences. We plan to integrate additional types of data into our system to support visualisation of functional MRI results, analysis of group-level statistics, and simulation of electric fields for neuromodulation. We are currently working to extend the visualisation capabilities by employing more advanced rendering techniques. We also plan to explore the use of new mechanisms for interacting with the system and data. We are particularly interested in incorporating sound, including audio streaming to enable communication with remote users, audio cues to indicate available or taken actions, and speech recognition for issuing commands to the VR system, including dictating annotations that would be attached to the data. We are also interested in the use of hand tracking and gesture recognition as another mode of control for the system. With the initial framework we have developed, we expect that we will be able to develop many of these new capabilities rapidly to provide a broader range of applications for our software.

5. Acknowledgments: Data were provided in part by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. Financial support for the Beijing Normal University data used in this project was provided by a grant from the National Natural Science Foundation of China: 30770594 and a grant from the National High Technology Program of China (863): 2008AA02Z405.

6. Funding and declaration of interests: Research reported in this publication was supported in part by the National Institute of Neurological Disorders and Stroke of the National Institutes of Health under Award Number R01NS074980. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

7. Conflict of interest: None declared.

8 References

[1] Zhang S., Demiralp Ç., DaSilva M., *et al.*: 'Toward application of virtual reality to visualization of DT-MRI volumes'. Proc. of the 4th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI '01, Berlin, Heidelberg, 2001, pp. 1406–1408

[2] Chen B., Moreland J., Zhang J.: 'Human brain functional MRI and DTI visualization with virtual reality', *Quant. Imaging. Med. Surg.*, 2011, **1**, pp. 11–16

[3] Ard T., Krum D.M., Phan T., *et al.*: 'NIVR: neuro imaging in virtual reality'. Proc. IEEE Virtual Reality (VR), Los Angeles, California, USA, 2017, pp. 465–466

[4] Duncan D., Garner R., Zrantchev I., *et al.*: 'Using virtual reality to improve performance and user experience in manual correction of MRI segmentation errors by non-experts', *J. Digit. Imaging*, 2018, pp. 1–8, doi: 10.1007/s10278-018-0108-5

[5] Egger J., Gall M., Wallner J., *et al.*: 'HTC vive MeVisLab integration via OpenVR for medical applications', *PLoS One*, 2017, **12**, p. e0173972

[6] Martin K.: 'Using virtual reality devices with VTK'. Available at: https://blog.kitware.com/using-virtual-reality-devices-with-vtk/, accessed: 2018-08-05

[7] Arikatla S., Fillion Robin J.C., Paniagua B., *et al.*: 'Bringing virtual reality to 3D Slicer'. Available at: https://blog.kitware.com/slicervirtualreality/, accessed: 2018-08-05

[8] Shattuck D.W., Leahy R.M.: 'BrainSuite: an automated cortical surface identification tool', *Med. Image Anal.*, 2002, **6**, pp. 129–142

[9] Joshi A.A., Shattuck D.W., Leahy R.M.: 'A method for automated cortical surface registration and labeling', in Dawant B.M., Christensen G.E., Fitzpatrick J.M., Rueckert D., (Eds.): 'Biomedical image registration' (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), pp. 180–189

[10] Haldar J.P., Leahy R.M.: 'Linear transforms for Fourier data on the sphere: application to high angular resolution diffusion MRI of the brain', *NeuroImage*, 2013, **71**, pp. 233–247

[11] Bhushan C., Haldar J.P., Choi S., *et al.*: 'Coregistration and distortion correction of diffusion and anatomical images based on inverse contrast normalization', *NeuroImage*, 2015, **115**, pp. 269–280

[12] Habibi A., Ilari B., Crimi K., *et al.*: 'An equal start: absence of group differences in cognitive, social, and neural measures prior to music or sports training in children', *Front. Hum. Neurosci.*, 2014, **8**, p. 690

[13] Phillips O.R., Joshi S.H., Squitieri F., *et al.*: 'Major superficial white matter abnormalities in Huntington's disease', *Front. Neurosci.*, 2016, **10**, p. 197

[14] MacKenzie Graham A., Lee E.F., Dinov I.D., *et al.*: 'A multimodal, multidimensional atlas of the C57BL/6J mouse brain', *J. Anat.*, 2004, **204**, pp. 93–102

[15] Shattuck D.W., Mirza M., Adisetiyo V., *et al.*: 'Construction of a 3D probabilistic atlas of human cortical structures', *NeuroImage*, 2008, **39**, pp. 1064–1080

[16] Wang R., Benner T., Sorensen A.G., *et al.*: 'Diffusion toolkit: a software package for diffusion imaging data processing and tractography'. Proc. Int. Soc. Mag. Reson. Med., Berlin, Germany, 2007, vol. 15, p. 3720

[17] Norton I., Essayed W.I., Zhang F., *et al.*: 'Slicerdmri: open source diffusion MRI software for brain cancer research', *Cancer Res.*, 2017, **77**, pp. e101–e103

[18] Fischl B.: 'Freesurfer', *NeuroImage*, 2012, **62**, pp. 774–781

[19] Rivière D., Geffroy D., Denghien I., *et al.*: 'BrainVISA: an extensible software environment for sharing multimodal neuroimaging data and processing tools'. Proc. 15th HBM, San Francisco, California, USA, 2009

[20] Swan J.E., Yagel R.: 'Slice-based volume rendering' (Ohio State University, Columbus, Ohio, USA, 1993). OSU-ACCAD-1/93-TR1

[21] Toro R.: 'Braincatalogue', 2018. Available from: https://braincatalogue.org/Bottlenose_dolphin, accessed June 10, 2018

[22] Le Bihan D., Johansen Berg H.: 'Diffusion MRI at 25: exploring brain tissue structure and function', *NeuroImage*, 2012, **61**, pp. 324–341

[23] Basser P.J., Mattiello J., LeBihan D.: 'MR diffusion tensor spectroscopy and imaging', *Biophys. J.*, 1994, **66**, pp. 259–267

[24] Mori S., Crain B.J., Chacko V.P., *et al.*: 'Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging', *Ann. Neurol.*, 1999, **45**, pp. 265–269

[25] Behrens T.E.J., Johansen Berg H., Woolrich M.W., *et al.*: 'Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging', *Nat. Neurosci.*, 2003, **6**, pp. 750–757

[26] Yeh F.C., Verstynen T.D., Wang Y., *et al.*: 'Deterministic diffusion fiber tracking improved by quantitative anisotropy', *PLoS One*, 2013, **8**, p. e80713

[27] Garyfallidis E., Brett M., Amirbekian B., *et al.*: 'Dipy, a library for the analysis of diffusion MRI data', *Front. Neuroinform.*, 2014, **8**, p. 8

[28] Van Essen D.C., Ugurbil K., Auerbach E., *et al.*: 'The human connectome project: a data acquisition perspective', *NeuroImage*, 2012, **62**, pp. 2222–2231