

Synchronized 2D/3D optical mapping for interactive exploration and real-time visualization of multi-function neurological images



Qi Zhang ^{a,b,*}, Murray Alexander ^{c,1}, Lawrence Ryner ^{d,2}

^a Centre for Imaging Technology Commercialization, London, Ontario, N6G 4X8 Canada

^b Department of Medical Biophysics, University of Western Ontario, London, Ontario, N6A 5C1 Canada

^c Department of Physics, University of Winnipeg, Winnipeg, Manitoba, R3B 2E9 Canada

^d Medical Physics, CancerCare Manitoba, Winnipeg, Manitoba, R3E 0V9 Canada

ARTICLE INFO

Article history:

Received 17 November 2012

Received in revised form 27 May 2013

Accepted 21 June 2013

Keywords:

Real-time
Visualization

Neurological image

Multi-function

Synchronization

Optical mapping

Lookup table

fMRI

Image fusion

Registration

Human-computer interaction

ABSTRACT

Efficient software with the ability to display multiple neurological image datasets simultaneously with full real-time interactivity is critical for brain disease diagnosis and image-guided planning. In this paper, we describe the creation and function of a new comprehensive software platform that integrates novel algorithms and functions for multiple medical image visualization, processing, and manipulation. We implement an opacity-adjustment algorithm to build 2D lookup tables for multiple slice image display and fusion, which achieves a better visual result than those of using VTK-based methods. We also develop a new real-time 2D and 3D data synchronization scheme for multi-function MR volume and slice image optical mapping and rendering simultaneously through using the same adjustment operation. All these methodologies are integrated into our software framework to provide users with an efficient tool for flexibly, intuitively, and rapidly exploring and analyzing the functional and anatomical MR neurological data. Finally, we validate our new techniques and software platform with visual analysis and task-specific user studies.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

A major goal of intracranial disease treatment is to achieve complete lesion resection while preserving normal brain tissues and functions, which is key to improve patients' quality of life and facilitate adjuvant therapies such as radiotherapy and chemotherapy [1]. However, despite the recent technology advancement and the progress of medical image understanding, the prognosis of some malignant brain lesions is still devastatingly poor, furthermore, the nature and structure of the brain and lesions usually lead to additional unpredictable surgical complications [2].

To achieve an optimal treatment, it is critically important to enhance the surgeon's ability of precisely locating

intracranial lesions and effectively identifying their spatial relationships with healthy tissues [3]. In this process, functional magnetic resonance imaging (fMRI), a technique of mapping human brain neuronal activity through measuring the brain's blood oxygen level-dependent (BOLD), is commonly used with the anatomical magnetic resonance (MR) images to describe the spatial distribution of nerves [4] and help neurosurgeons to effectively avoid eloquent areas of the brain during surgical intervention process [5,6].

1.1. Existing software packages

There have been some software products that can be used in the medical application domain such as brain disorder diagnosis and treatment. For example, MRICro [7] is a freeware for displaying slices of anatomical MR data along with functional overlays on nontransparent volumetric brain images. ParaView [8] is an open source package built on the top of the Visualization Tool Kit (VTK) [9] libraries for data parallel computing and rendering on distributed systems. 3D Slicer [10] is another VTK-based open-source software package for image analysis, processing and display, and is extensible to integrating new image processing routines into

* Corresponding author at: Centre for Imaging Technology Commercialization, London, Ontario, N6G 4X8 Canada. Tel.: +1 519 858 5013x24294; fax: +1 519 931 5719.

E-mail addresses: qzhang58@uwo.ca, qi.zhang16@gmail.com (Q. Zhang), mu.alexander@uwinnipeg.ca (M. Alexander), Lawrence.Ryner@cancercare.mb.ca (L. Ryner).

¹ Tel.: +1 204 258 2981; fax: +1 204 774 4134.

² Tel.: +1 204 787 8537; fax: +1 204 984 7036.

its working pipeline. DataViewer3D [11] is a program depicting multimodality neuroimaging data that is built on VTK and Python and supports two-dimensional (2D) and three-dimensional (3D) data projections. There are also commercial software available for visualizing biomedical data acquired from multiple sources and modalities, such as Amira [12]. Recently, a web-based interactive medical image viewer was developed by Mahmoudi et al. [13] to remotely explore neurological data, and VTK and Insight Segmentation and Registration Toolkit (ITK) [14] libraries were intensively used for image exploration and processing.

Software packages have also been developed for statistically analyzing neurological data and extracting relevant brain activity information from blood flow changes. For example, BrainVoyager QX [15] is a commercial software for analyzing functional and structural MR data. SPM [16] is a freely available package designed to process brain image sequences, which is implemented with MATLAB and newly developed toolboxes can be integrated into its working environment; for example, "BrainBlend" [17] is one of such toolboxes, and works with the software "Blender" [18] to show the relevant BOLD changes. FSL [19] is another software platform for multi-function brain image analysis, and a VTK-based software, FSLView [20], is available to visualize the analysis results. In addition, AFNI [21] is a set of expandable C programs for processing and displaying fMRI data defined over 3D volumes and 2D cortical surface meshes.

1.2. Visualization algorithms

Even though the described software packages play an important role in neurological disease diagnosis and treatment, the lack of some valuable features, the low rendering performance, and the difficulties of data manipulation and diagnostic information extraction in large data visualization usually impede their wide clinical applications.

At the same time, there are a wealth of academic publications on medical data rendering and exploration. For example, Rößler et al. [22] presented a graphics processing unit (GPU)-based technique for displaying functional and anatomical brain images. To provide users with multiple imaging information, Beyer and his colleagues implemented a multi-volume raycasting algorithm running on graphics hardware [23], and a similar technology was developed by Köhn's research group to visualize segmented anatomical and functional MR images [24]. Furthermore, Rieder and his colleagues [25] integrated a distance-based enhancement technique into a medical data display framework to emphasize both fMRIs and lesions, and this group also introduced a volume clipping algorithm [26] for exploring multi-spectral MR images and identifying inhomogeneous pathological tissues. Recently, to explore complex anatomical structures, Herghelegiu [27] developed an algorithm to render segmented structures from T₁ contrast enhanced MR data and T₂ weighted MR sequence simultaneously, while Diepenbroc et al. [28] presented a medical data enhancement method to provide clinicians with various imaging information during treatment planning process.

Transfer function (TF) is critically important in changing the appearance of the displayed volume and visualizing structure of interest. Selvera et al. [29] designed a TF to improve interactive volumetric data exploration, and Pinto et al. [30] presented techniques of combining two-level TF design for direct volume rendering.

1.3. Current challenges

As described in Sections 1.1 and 1.2, most available softwares for medical data exploration are built on the top of either VTK and ITK libraries [8,10,11,13], or the standalone compiled language such as C and C++ [15,19,21,7,12], or the script language such as

MATLAB [16–18]. Usually, they may have comprehensive functions, however, their abilities of keeping up with the technology development, visualizing large size medical data in real time, and manipulating the displayed data in a user-friendly way are weak. For instance, a commonly used routine for exploring neurological data is to fuse anatomical and functional brain images, highlight the activated fMRI voxels and the anatomical structures of interest, and navigate the rendered 3D image with 2D slices. However, in the commonly used software platforms, the color and opacity adjustment and mapping for 3D volume rendering and 2D slice display are independent. Due to the independence of 3D and 2D color and opacity mapping, the users have to first use a TF to adjust the rendered volume and capture the tissue of interest, and then they have to manipulate the lookup table and window-level to show the multi-layer 2D slice images. This is a series of time consuming operations, and when there are a set of images needed to be adjusted and multiple features needed to be captured and updated, it is a challenging task to get satisfactory results within a restricted time span.

On the other hand, some published medical data visualization methodologies such as those described in Section 1.2 usually mainly concentrate on algorithm development, not paying enough attention to integrating the newly developed techniques into the commonly used medical image libraries such as VTK and ITK, so it is still difficult to use them within the medical community. Furthermore, they usually have not efficiently addressed the critical issues of applying the research results to improve the performance of comprehensive medical data rendering and exploration software platforms that can be used in daily medical applications.

All these shortcomings may slow down the progress of applying new real-time medical imaging technologies and software to improve the image-guided neurological disease diagnosis and treatment in real-world clinical practice.

1.4. Our methods

To address some of the challenges illustrated in Section 1.3, we developed algorithms and a software platform for multi-function neurological image real-time visualization, exploration, and 2D/3D synchronization. The software package was designed with an object-oriented modular structure, which is transparent and extensible to new functions, algorithms, and working pipelines.

Our artifact-free interactive voxel classification algorithm [31] and graphics hardware accelerated raycasting methodology [32] were integrated into the software framework for medical data real-time volume rendering and interactive manipulation, delivering high-quality fusion images in real time. To improve the multiple 2D slice image real-time fusion and enhance the display result, a novel opacity-adjustment algorithm was introduced to build optical lookup tables. Most importantly, we developed a new synchronized color and opacity mapping scheme for 2D and 3D medical image visualization, interactive adjustment, and real-time exploration through unified operations, which greatly improved multi-function neurological image manipulation efficiency and display accuracy.

All the newly developed algorithms and working pipelines were integrated into our comprehensive software package and optimized to work with the VTK and ITK libraries seamlessly. The data stream of the developed software framework was compatible with the commonly used neurological image processing and analysis software packages, providing an integrated pipeline for medical data process, analysis, and visualization. Finally, we consider that the described work in this paper as a new contribution to the currently developed medical imaging technologies and software.

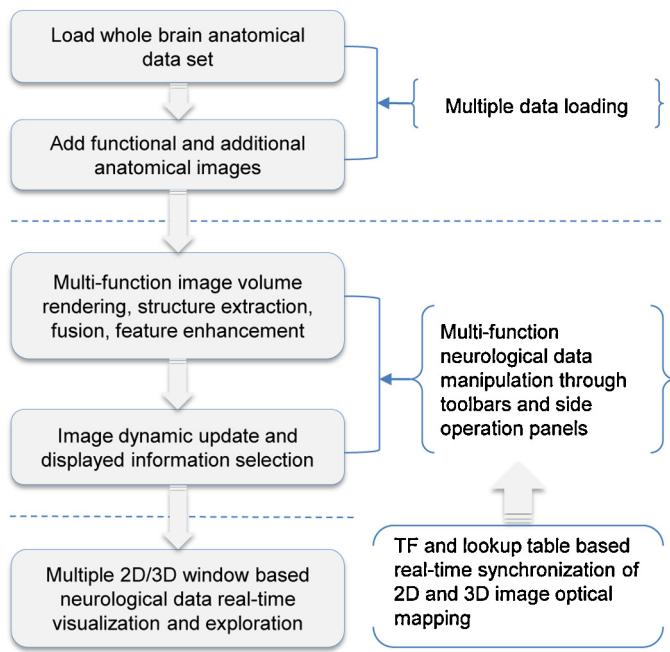


Fig. 1. Flowchart describing the software platform's working pipeline and architecture: loading and adding data, fusion, information extraction, feature enhancement, updating, visualization, and manipulation.

2. Platform architecture

This software platform was developed with C++, OpenGL [33], and OpenGL Shading Language (GLSL) [34], as well as VTK and ITK. The system's graphical user interface (GUI) was implemented with C++-based open-source libraries such as Qt [35], CTK [36], and MITK [37], connecting user-computer's interactions with the system's core functions for multi-function medical data real-time display and exploration. All the GUI components acquired from CTK and MITK can be integrated into the Qt-based GUI software programming pipeline, which is a straightforward implementation.

The software architecture was designed with an object-oriented functional layer based structure. The outer layer includes loading the main anatomical data, adding functional and additional anatomical images, and integrating image processing and statistical analysis results; the second constitutes extracting structures of interest and fusing multi-function images; the third consists of visualizing the processed images in multiple display windows for multi-view based image navigation; the fourth comprises the synchronization of the 2D/3D optical mapping and adjustment of the displayed images and enhancing features of interest; while the inner layer includes various high-performance image computation, processing, and volume rendering kernels.

The volume visualization and multimodality image fusion were implemented in the class *nrcOpenGLGPUVolumeRayCastMapper*, in which the ray casting kernel was running on GPU fragment shaders. It is a subclass of *vtkGPUVolumeRayCastMapper* and works with the class *nrcTransferFunction*, a subclass of *itk::Object*, for synchronized 2D/3D optical mapping and multimodality neurological image fusion and display. The programming pipelines of the new classes are the same as the working pipelines of VTK and ITK.

The system's main manipulation functions and user-software interactions were implemented through two side panels “Volume Manager” and “Image Inspector” as shown in Fig. 6. Fig. 1 depicts the flowchart of the software platform's architecture.

3. Main functions

The software package was designed for multi-function neurological data visualization and exploration, providing an intuitive user-system interaction and the realistic representation of the anatomical and functional image information. It has most of the common and advanced functions for medical image display, manipulation, and processing. Brain anatomies were reconstructed from MR data and functional images were extracted from statistical data processing, and the mutual information based registration algorithm was used to fuse different data sets.

The system's GUI includes a menu bar, a tool bar, and several side panels for controlling the image navigation, manipulation, and management. For each display window, its dimension can be freely updated and rendered images can be dragged and dropped between different image rendering windows. The cross line in the three 2D display windows can be dynamically updated and synchronized with the three orthogonal cross-volume planes in the volume rendering window, and the window-level of the slice images displayed in the 2D rendering windows and three cross-volume planes can be interactively adjusted. Furthermore, the image fusion and optical adjustment of the multi-function medical volumes can be real-time synchronized with the 2D slice image display. In the following, we briefly describe these main functions, which are implemented in the software manipulation panels, i.e., “Image Inspector” and “Volume Manager”. Their appearances and functions are illustrated in Figs. 2–5.

3.1. Image inspector

As shown in Fig. 2, the image inspector editor consists of two subeditors. The first is named “Images”, which manages all of the loaded data sets, and each of them can be adjusted through the cross-volume planes with sagittal (Sa), coronal (Co) and transversal (Tr) orientations separately. The second subeditor is “Information”, which displays the name of the selected image and its dimension, size, origin, as well as the pixel spacing and total pixel number. In this subeditor, the focus point information (pixel value and the distance from the image origin point) of the 2D cross-line can be real-time updated during the data navigation process.

3.2. Volume manager

This panel manages the loaded multi-function medical data sets, including volume generation, navigation, and manipulation. It can be used by various images through changing its parameter settings corresponding to different images by selecting the image names in the “Images” subeditor.

3.2.1. Navigation

This subpanel provides various functions for data navigation. It manages the window-level adjustment along with color and opacity mapping for cross-volume navigation planes, the corresponding slice images, and the rendered volumes. The data management includes the current main image and the additional fused multi-function images, all of which are controlled by the “Image Inspector” panel. In addition, this subpanel provides functions such as lookup table threshold setting, two optical mapping modes, i.e., 2D/3D independent mapping or synchronized mapping, as well as the display modes to customize medical data display and exploration. Fig. 3 illustrates the “Navigation” subpanel as well as the parameter management working pipelines for data mapping between different subpanels.

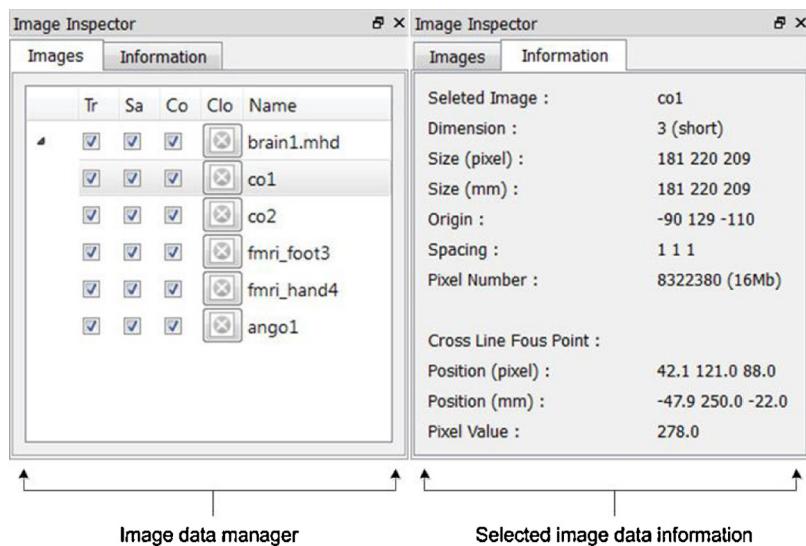


Fig. 2. Diagram describing the “Image Inspector” panel for data management and the display of image information and real-time updated data navigation: data management (left) and selected image information and navigation (right).

3.2.2. Generation

This subpanel takes charge of the volume rendering, optical mapping, structure enhancement, and rendering algorithm selection. TF editors are used to control the appearance of the rendered volumes. Different volume rendering algorithms can be selected and adjusted through a TF editor, i.e., GPU-based raycasting, software-based raycasting, 2D/3D texture mapping, as well as maximum intensity projection (MIP). Furthermore, shading enhancement can be added to each of the rendered medical volumes independently.

To save medical volume adjustment time and improve data navigation efficiency, the adjusted TFs with corresponding shading information can be saved to an Extensible Markup Language (XML) file, which is used to encode all of the adjusted parameters. The XML file can be loaded directly next time to integrate the previously saved rendering parameters to the system’s volume visualization pipeline, so the saved results can be restored automatically. Most importantly, the 3D optical mapping can be synchronized with the lookup table for the slice image display in real time, which will be described in Section 3.2.3. Fig. 4 illustrates the data set and the

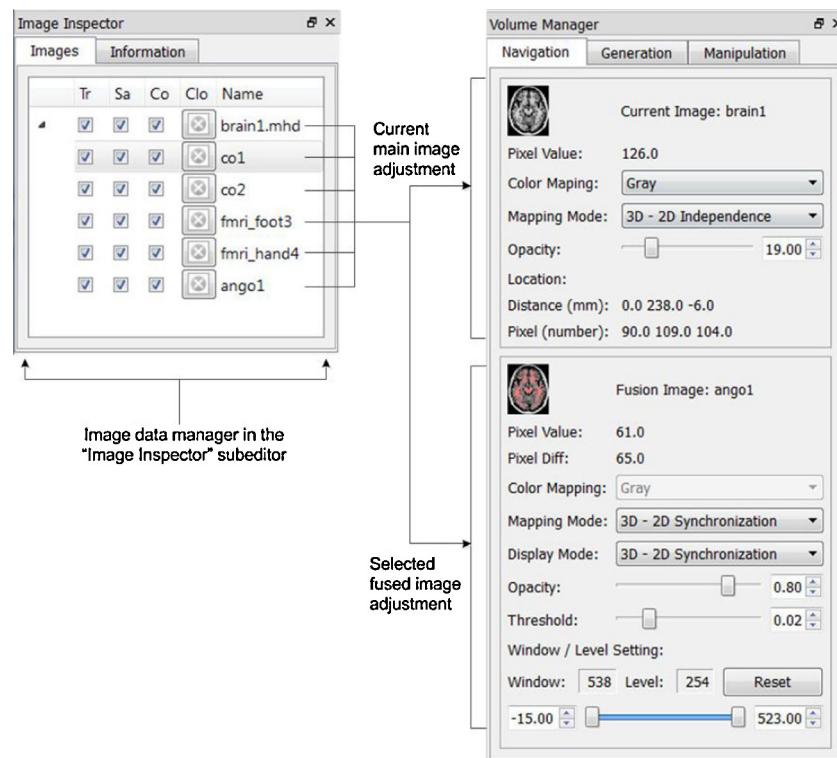


Fig. 3. Diagram depicting the mapping scheme from the data set management to data navigation subpanel, i.e., one image management subpanel can be used by multiple data sets through selecting different image name and then the subpanel parameters will be updated correspondingly.

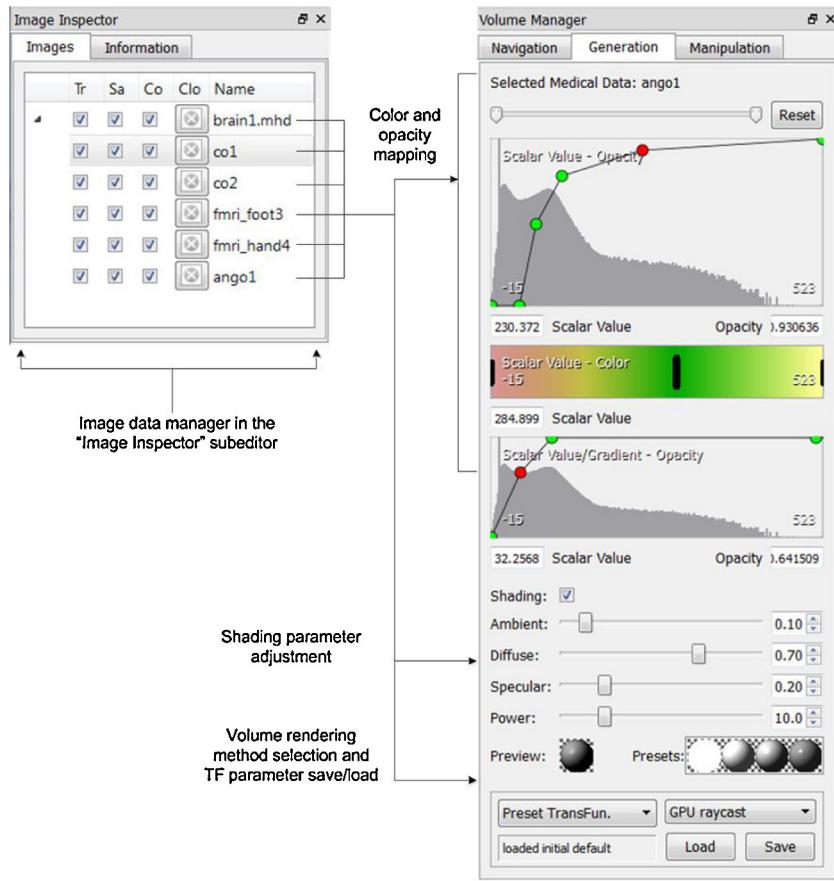


Fig. 4. The description of the volumetric medical image generation and adjustment subpanel, as well as the mapping scheme from the data set management to volume generation subpanel, i.e., one subpanel can be used by multiple data sets.

"Generation" subpanel, i.e., one editor can be used by multiple data sets through updating its parameters.

3.2.3. Manipulation

As described in Fig. 5, this subpanel includes the mapping mode selection, which determines whether the color and opacity of the three cross-volume planes are set independently or synchronically. Furthermore, the color and opacity of the cross-volume planes can be adjusted flexibly and can be synchronized with the color setting

of the three corresponding slice images displayed in the 2D image rendering windows.

3.3. Visualization

An important function of this software platform is the medical image display. The main volumetric anatomical data and the additional multi-function medical images can be fused and rendered in real-time, enhancing the complex 3D shapes and structures extracted from the loaded data sets. All the visualized images in

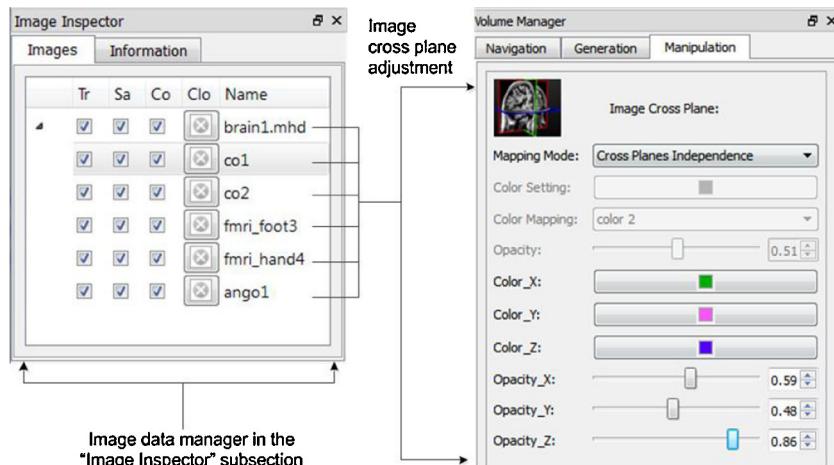


Fig. 5. Diagram describing the color and opacity mapping adjustment in the "Manipulation" subpanel as well as the image data management mapping.

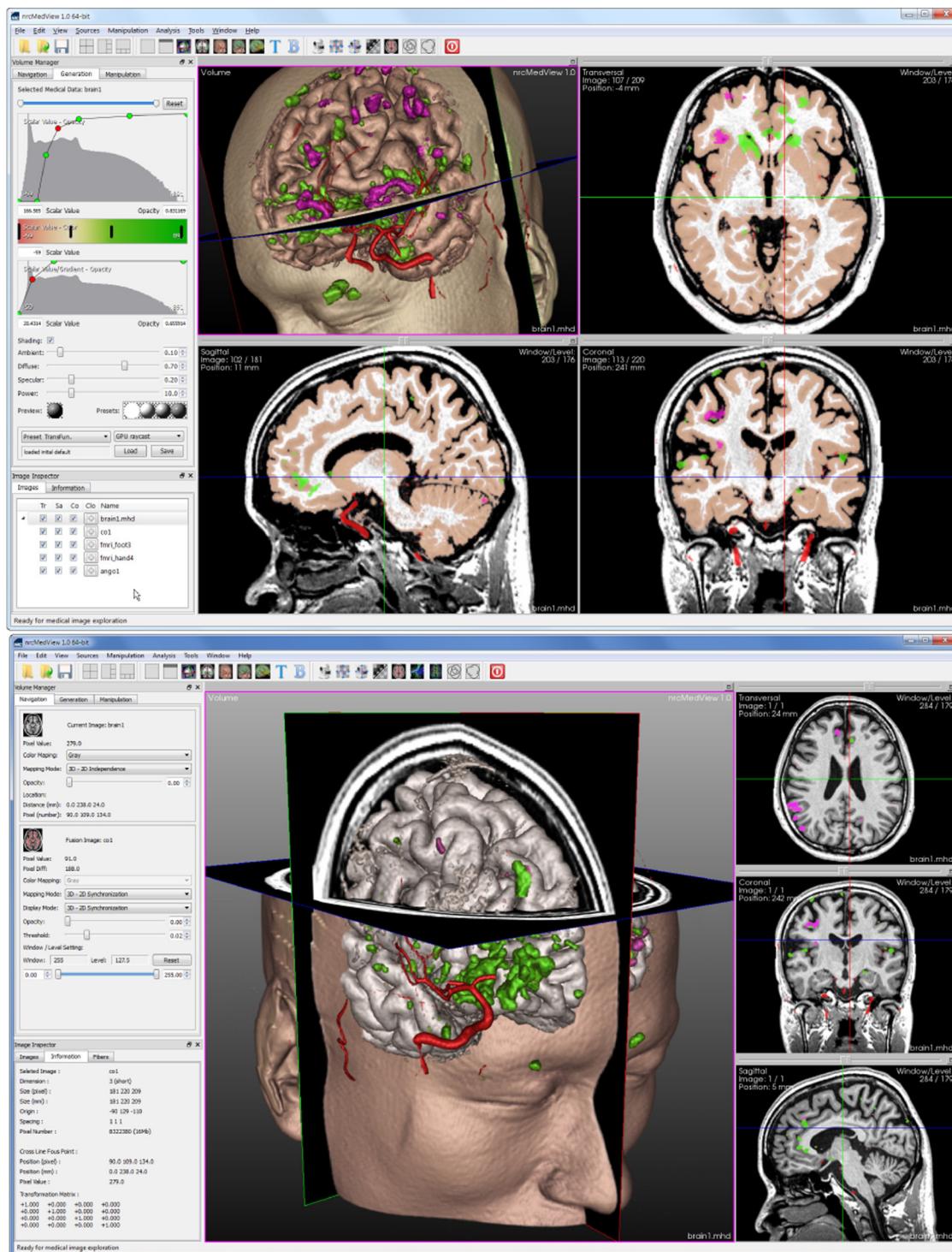


Fig. 6. Snapshot images describing three whole GUI structures of whole software platform. Multi-function neurological images are visualized, fused, and enhanced.

the 3D rendering window can be interactively navigated with up to three cross-volume planes, and the image information at the navigation locations is real-time displayed in the slice images in the corresponding 2D rendering windows.

The system's visualization module includes all the commonly used functions, such as clipping, color and opacity adjustment, 3D and 2D optical mapping and synchronization, image manipulation, cross-volume navigation, image information extraction and display, and structure of interest enhancement. Each of the three 2D slice image rendering windows has a slider at the top of the

window for the navigation of the displayed images slice by slice. To maximize the display area, the three image navigation sliders can be hidden by a toggle button.

GPU-based raycasting algorithm is used as the main volume rendering algorithm in this software platform. To facilitate the volumetric data display and manipulation, this system also provides users several additional visualization methodologies as optional choices. Fig. 6 illustrates three snapshots of the software platform, where the entire MR anatomical brain image, segmented grey matter, blood vessels, and two fMRI images of hand and foot motor task

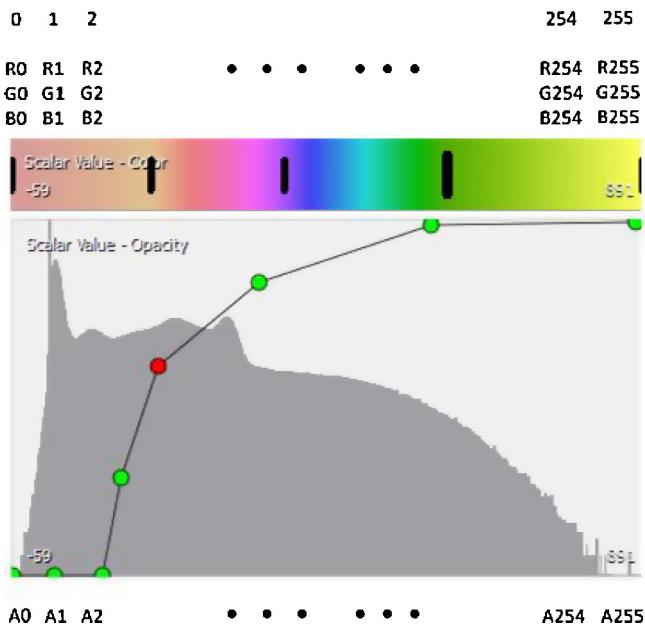


Fig. 7. Diagram describing the color (top) and opacity (bottom) lookup table: mapping intensity value to color and opacity.

are displayed, fused and enhanced with different volume rendering and optical mapping parameters and window layouts.

4. Optical mapping and synchronization

In this section, we first describe the techniques for multiple medical volume visualization with artifact-free voxel classification algorithms integrated, then methodologies for slice image display and fusion are presented and analyzed. Finally, we illustrate a real-time scheme for synchronized 2D and 3D medical image optical mapping and exploration.

4.1. Multi-volume rendering

The platform first loads the medical data sets to its memory and then GPU-based raycasting or texture mapping algorithm is used to volume render the loaded images. The first loaded image is used as the main volume, storing the entire brain anatomical structure, while the data sets loaded later are used as fusion images, such as fMRIs, blood vessels, and grey matter.

First, when a data set is loaded, assume its voxel intensity range is $[v_{\min}, v_{\max}]$. For every voxel v_x in this range, i.e., $v_x \in [v_{\min}, v_{\max}]$, Eq. (1) is used to calculate the new mapped voxel value \tilde{v}_x .

$$\tilde{v}_x = \frac{v_x - v_{\min}}{v_{\max} - v_{\min}} \times 255 \quad (1)$$

Next, the lookup table as shown in Fig. 7 is used to calculate the mapped color (R_x, G_x, B_x) and opacity A_x , when \tilde{v}_x is not an integer from 0 to 255, linear interpolation is used to calculate its mapped color and opacity. During the volume rendering process, our artifact-free post-color attenuated classification algorithm [31] is employed to build the TF that is used for color and opacity mapping: for every two adjacent points $\mathbf{r}(t_i)$ and $\mathbf{r}(t_{i+1})$ on the casting ray $\mathbf{r}(t)$, assume their initial intensity values are s_f and s_b respectively, which are then mapped to the range $[0, 255]$ using Eq. (1). Assuming the segment length is d and the casting ray is uniformly sampled, and $\tau(s)$ is the transparency of the voxel with intensity value s , i.e., $\tau(s) = 1 - \alpha(s)$, where $\alpha(s)$ and $c(s)$ are the mapped opacity and color of a voxel with intensity s . The opacity and color mapping pipeline is based on the lookup table described in Fig. 7. The

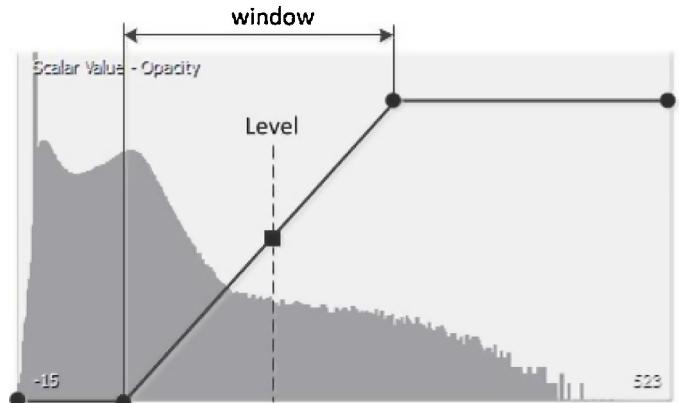


Fig. 8. Window and level as well as slice histogram used to build lookup table for 2D slice display.

post-color attenuated classification algorithm expressed by the Eqs. (2) and (3) is used to compute the mapped color $C = C(s_f, s_b, d)$ and opacity $\alpha = \alpha(s_f, s_b, d)$, which are then employed to build the color and opacity lookup table, and the Riemann sum is used to calculate the color and opacity integral. The concrete implementation of using the artifact-free classification algorithm to build lookup table for color and opacity mapping in volume rendering computation is illustrated in Ref. [38].

$$C \approx \left[1 - \frac{1}{2} \alpha(s_f, s_b, d) \right] \times \frac{d}{s_b - s_f} \left(\int_0^{s_b} \tau(\mu) c(\mu) d\mu - \int_0^{s_f} \tau(\mu) c(\mu) d\mu \right) \quad (2)$$

$$\alpha \approx 1 - \exp \left[-\frac{d}{s_b - s_f} \times \left(\int_0^{s_b} \tau(\mu) d\mu - \int_0^{s_f} \tau(\mu) d\mu \right) \right] \quad (3)$$

Finally, the calculated color C and opacity α are used in the volume rendering computation. In the case of multiple volumes being loaded, the algorithm will map color and opacity for each of them separately, and the computed volume rendering results will be fused together in real time for multiple medical volume visualization.

4.2. Slice image display and fusion

Color and opacity lookup table is commonly used in the process of displaying and fusing multiple 2D images. Currently, most available softwares usually build a lookup table using a VTK-based 2D image window-level, as shown in Fig. 8.

The range of the built lookup table is $[0, 255]$, and Eq. (1) is used to map a pixel's intensity to a value in the range $[0, 255]$. Bilinear interpolation is used to map a pixel value in the range $[0, 255]$ to color and opacity. However, due to VTK does not map a pixel with zero intensity to an optical value with totally transparent opacity, so when displaying multiple 2D images, the final fused image will be dark and the underlayer images cannot be viewed clearly. The experimental result demonstrating this effect will be shown in Section 6.

To address the zero-intensity opacity problem, a commonly used method is setting a threshold λ in the process of building the VTK-based lookup table. For every pixel in the selected slice, first map its value to the range $[0, 255]$ using the Eq. (1), then check whether the mapped value is less than the preset threshold λ . If it is true, map its opacity to zero; otherwise, do the normal color and opacity mapping as described previously. Using this

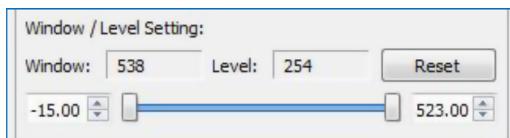


Fig. 9. The editor and corresponding tools for window and level setting, adjustment, and display.

method, the clearness of the final overlaid images is improved, however, some important features whose intensities are lower than the preset threshold λ become invisible in the final fused images.

We therefore proposed an opacity-adjustment algorithm to address the described issues. First, build a lookup table with 256 indices from 0 to 255 according to the VTK working pipeline. For every index k ($k=0, 1, \dots, 255$), calculate the largest value of its three color elements, i.e., red (R_k), green (G_k), and blue (B_k). Assume the calculated value is A_k , i.e., $A_k = \max(R_k, G_k, B_k)$, normalize it to the range [0, 1], i.e., $\tilde{A}_k \in [0, 1]$, which is then used as a new opacity. Next, create a lookup table using the original color (R_k, G_k, B_k) and the new opacity \tilde{A}_k , so when the mapped color is black, i.e., $R_k = G_k = B_k = 0$, then $\tilde{A}_k = 0$, which means that the mapped pixel is totally transparent. Furthermore, when the mapped color is close to black, i.e., $\Theta_k < \varepsilon$ (ε is a positive threshold close to zero, $\Theta = R, G$, or B), then its opacity is also close to zero, which means that the mapped pixel is close to be transparent. Therefore, when multiple image slices are fused together, our algorithm can guarantee that the important features of the overlaid images are shown clearly in the final displayed images.

One shortcoming of our method is that when the intensity range of a structure of interest (SOI) is narrow, such as the angiography image, it is difficult to extract the SOI from the data for correct visualization. Furthermore, due to the lack of information about the histogram of every slice image that corresponds to the SOI, it is a time consuming operation to set an optimized window and level for building the 2D lookup table. We therefore presented a new synchronized 3D and 2D color and opacity mapping strategy to address this issue, the will be illustrated in the following [Section 4.3](#).

4.3. Synchronization

The synchronization algorithm of the color and opacity real-time mapping for both 2D slice and 3D volume visualization is listed in the following items.

– Assume there are N fusion data sets Ω_i ($i=0, 1, \dots, N-1$). For every loaded data set Ω_i , first calculate its maximum and minimum value v_{\max}^i and v_{\min}^i , and then the Eqs. (4) and (5) are used to compute the corresponding window and level, i.e., v_{win}^i and v_{lev}^i , of the data set Ω_i . The calculated window and level v_{win}^i and v_{lev}^i can be further adjusted using the window-level editor Υ_i , which is demonstrated in [Fig. 9](#)

$$v_{\text{win}}^i = v_{\max}^i - v_{\min}^i \quad (4)$$

$$v_{\text{lev}}^i = \frac{v_{\min}^i + v_{\text{win}}^i}{2.0} \quad (5)$$

– For every loaded fusion image data Ω_i , there is a corresponding TF T_i , which can be adjusted through the TF editor Γ_i described in [Fig. 7](#). The T_i includes color and opacity mapping rules that will be employed to build the volume rendering lookup table L_i . – Integrate the calculated and adjusted window and level, i.e., v_{win}^i and v_{lev}^i , of the fusion image data Ω_i into their corresponding TF

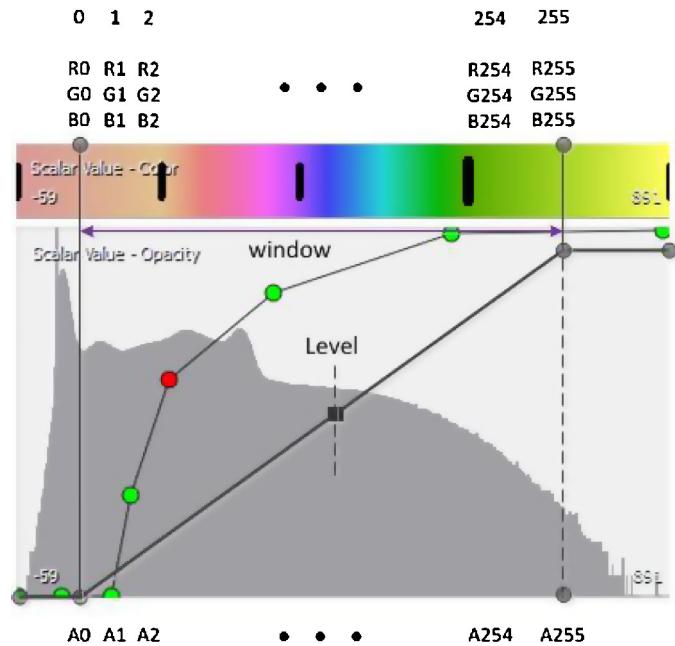


Fig. 10. Diagram describing the new color mapping pipeline: combining the TF T_i and TF editor Γ_i with the calculated window v_{win}^i and level v_{lev}^i . The lookup table \hat{L}_i is built between the window range with mapping rule determined by the control points of Γ_i . For every point between the controlled points (green or red), linear interpolation is used to calculate its mapped optical value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

editor Γ_i , and use this combination to build a new color and opacity lookup table \hat{L}_i , where the mapping indices are from 0 to 255 that are uniformly spread in the range of the window v_{win}^i and centered at the level v_{lev}^i . Linear interpolation will be used to calculate the index values when they are not on the control points, i.e., the red and green points in [Fig. 10](#). The actual lookup table \hat{L}_i mapping rule is determined by two factors. The first factor is TF T_i , which is controlled through adjusting Γ_i , and the second one is the window v_{win}^i and level v_{lev}^i , which can be adjusted through window-level editor Υ_i ($i=0, 1, \dots, N-1$).

- In the newly built lookup table \hat{L}_i , all the control points, i.e., green and red points in [Fig. 10](#), can be dynamically updated, i.e., changing their locations, adding new control points by double click the left mouse button, or deleting existing control points by click right mouse button on the control point that will be deleted.
- Synchronization: when adjusting and updating the control points in the TF editor Γ_i , the software platform will real-time update the TF lookup table \hat{L}_i as shown in [Fig. 7](#), where the Eqs. (2) and (3) are automatically used to calculate the corresponding color and opacity values during volume rendering process, so the transparency and color of the volume rendered images will be updated in real-time. At the same time, these updated control points are used to build the 2D color and opacity lookup table, as shown in [Fig. 10](#), which is in the range [0, 255] and can be directly used to set the color and opacity of the slice images displayed in the 2D rendering windows. The overall flowchart is illustrated in [Fig. 11](#).

Through using the synchronized color and opacity mapping methodology, the users can adjust multi-function medical data visualization and fusion, enhance the structure of interest, and capture features of interest in both 2D and 3D images simultaneously. The synchronization scheme is real time, when users adjust the appearance of the volume rendered images, their adjustments will be instantly used to update the color and opacity mapping of the

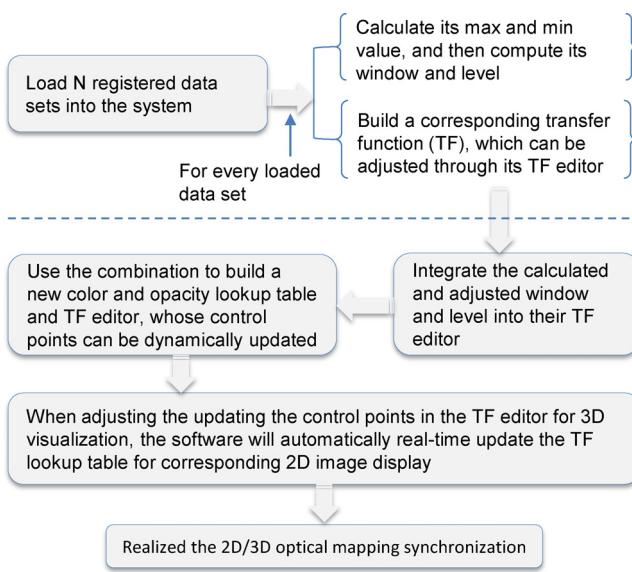


Fig. 11. The flowchart of the 2D and 3D optical mapping synchronization for multimodality neurological image exploration and manipulation.

corresponding 2D slice images. The new synchronized 2D and 3D optical mapping technique will improve the efficiency of the multi-function medical data exploration and navigation.

Fig. 12 describes the working pipeline our 2D/3D optical mapping synchronization algorithm. In this example, we have three fusion data sets, i.e., Ω_0 (blood vessel acquired from angiographic image), Ω_1 (brain white matter), and Ω_2 (function MR image, i.e., fMRI of hand stimulus). These three data sets are optically mapped through their transfer functions T_i and adjusted through corresponding TF editors $\Gamma_i(i=0, 1, 2)$. For every Γ_i , we have two adjustments with corresponding TFs T_i^0 and $T_i^1(i=0, 1, 2)$ respectively. The window-level editors $\Upsilon_i(i=0, 1, 2)$ are used to adjust the TF's window and level. Here, only show one window-level editor, i.e., Υ_1 (corresponding to white matter Ω_1), and there are two adjustments, i.e., Υ_1^0 and Υ_1^1 .

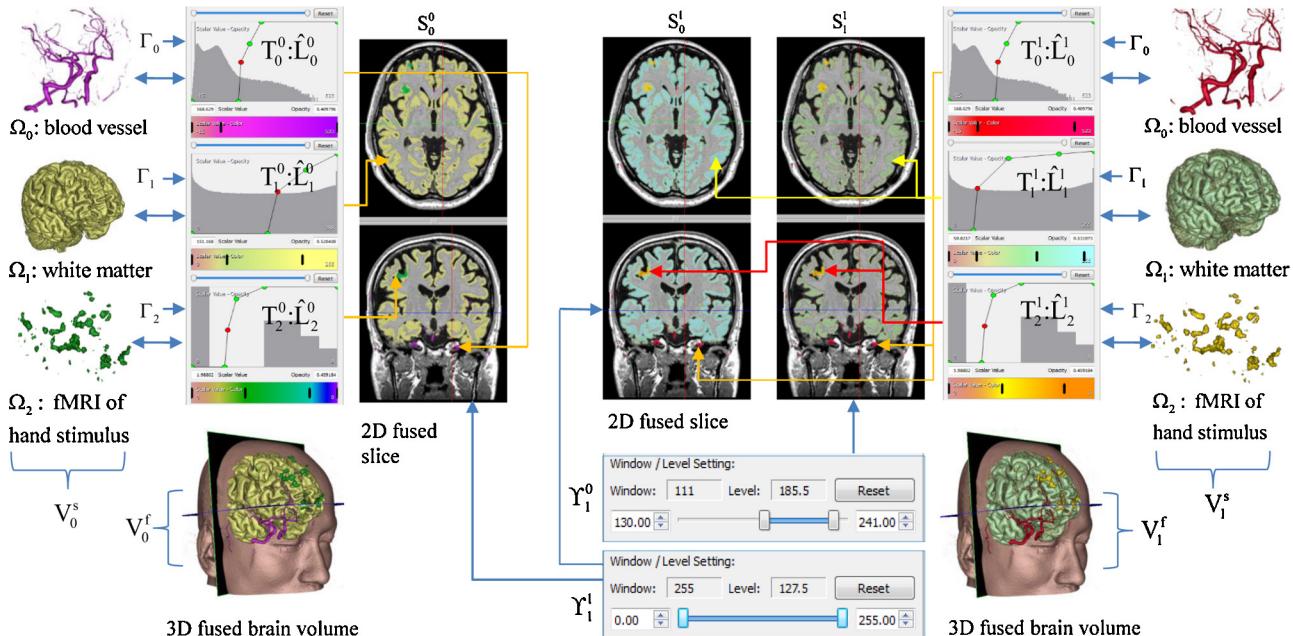


Fig. 12. Diagram describing the working pipeline of the 2D and 3D neurological image synchronization for color and opacity mapping and adjustment with three fused data sets used, i.e., blood vessel, white matter and fMRI of hand stimulus.

For every TF $T_i(i=0, 1, 2)$, Eqs. (2) and (3) are used to build lookup table L_i . The widow and level, being adjusted through Υ_i , are used to create new 2D/3D synchronized lookup table \hat{L}_i . When adjusting the TF T_i^0 through TF editor Γ_i , we can do color and opacity mapping to visualize Ω_i , creating 3D volumes V_0^s (including three volumes, blood vessel, white matter and fMRI) and the fused volume V_0^f . At the same time, our adjustments, i.e., $T_i(i=0, 1, 2)$, are automatically used to map color and opacity for 2D slice display of blood vessel (Ω_0), white matter (Ω_1) and fMRI (Ω_2), which are included in S_0^0 .

When performing the second TF adjustment, the system generates V_1^s, V_1^f and $S_1^j(j=0, 1)$. The white matter in S_0^1 and S_1^1 are generated with the same TF adjustment T_1^1 and different window/level, which are 255/127.5 and 111/185.5 respectively. We can see the unified and synchronized color and opacity mapping in both 2D slice and 3D volume in Fig. 12. In addition, through setting different window and level Υ_1^0 and Υ_1^1 , we can enhance the 2D visual result of the corresponding data set, here optical mapping of white matter Ω_1 is adjusted through manipulating window/level editor Υ_1 , whose value settings are shown in Υ_1^1 and Υ_1^0 . Here Υ_1^1 is used to generate S_0^0 and S_1^1 , while Υ_1^0 is employed to create S_1^0 .

5. Image processing and analysis

The neurological data sets used in the paper were acquired with a Siemens Trio Tim 3T MRI Scanner. The acquired data include T_1 - and T_2 -weighted anatomical images providing structural reference, BOLD fMRI (Blood Oxygen Level Dependent functional MRI) of hand and foot motor tasks for identifying activation areas, as well as MR angiography data for extracting vessel structures. DCM2NII [39] is used to convert the image format from DICOM to NIfTI for statistical analysis. In this section, we briefly describe the data acquisition, processing and statistical analysis.

5.1. Data acquisition

Anatomical MR brain images are first required, i.e., T_1 - and T_2 -weighted brain scans. The intra-plane image resolution is 256×256 with pixel size $1.0 \text{ mm} \times 1.0 \text{ mm}$. 176 slice images are acquired with

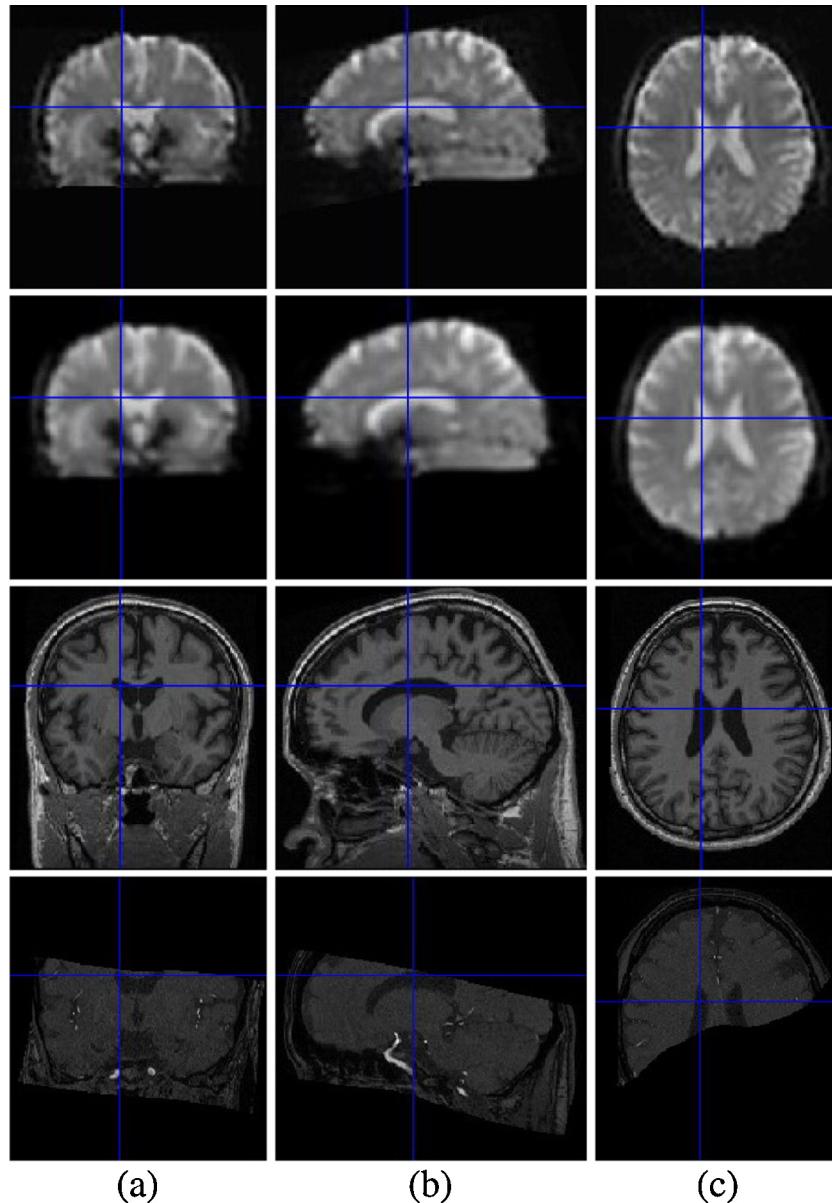


Fig. 13. Normalized and realigned time-series BOLD fMRI images of foot motor task (first row) and smoothed with a Gaussian kernel (second row); the normalized and registered and transformed anatomical brain image (third row); and the registered and transformed angiography image (fourth row). Slice orientation: column (a) coronal; (b) sagittal, and (c) axial (transverse).

thickness 1 mm, repetition time (TR) 1.9 s and echo time (TE) 2.2 s. The next acquired data are the two time-series BOLD fMRI images for stimulus of foot and finger motion with flip angle 90°, both of which have intra-plane image resolution 384 × 384 with pixel size 3.75 mm × 3.75 mm, and 120 slices are acquired with thickness 4 mm. The TR and TE of the acquired image are 2.0 s and TE 40 s respectively. The final acquired data is angiography MR image. The intra-plane image resolution is 384 × 320 with pixel size 0.52 mm × 0.52 mm. Total 128 slices are acquired with slice thickness is 0.7 mm and the flip angle 15°.

5.2. Image processing

In this section and the following Section 5.3 we will illustrate the processing and statistical analysis of the acquired neurological images. The open source software package SPM [16] is used in this process.

5.2.1. Realignment

To eliminate movement artefact, the two set of time-series BOLD fMRI images are first realigned, i.e., the first scanned image of the foot motor task fMRI series is used as a reference, and the first image of the hand motor task fMRI series is realigned to it. Then, the rest of images are realigned to the first scanned image in each of these two groups and a mean image is created in the first image group.

Algorithms of least-squares and rigid body spatial transformation are used for image transformation, and a B-Spline interpolation is employed to reslice these two groups of images to make them match to the first scanned image voxel-for-voxel.

5.2.2. Coregistration

The T₁- and T₂-weighted anatomical images are first registered to the generated mean image, then the angiography image is registered to T₁-weighted image. The registration involves finding parameters to maximize a normalized mutual information based objective function. To smooth the cost function and decrease the

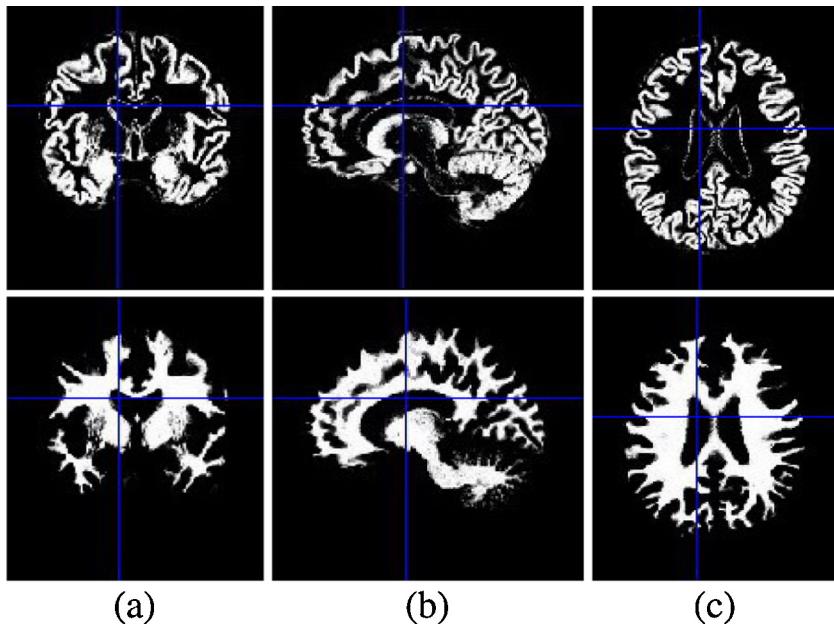


Fig. 14. The segmented and registered grey matter (second row) and grey matter (third row). Slice orientations are the same as those in Fig. 13.

chance of local minima, a Gaussian filter with 7 mm smoothing kernel is applied to the 256×256 joint histogram.

5.2.3. Segmentation and normalization

The working pipeline includes a two-step computation: the first is affine regularization, i.e., the initial starting images are placed in approximate alignment with the tissue probability maps (TPMs) to avoid local extremum. The second is normalization, i.e., determining a 12-parameter affine transformation, during which the registration is initially performed through matching the whole brain to a template, and then is proceeded by only matching the brains together through template voxel weighting.

The above pipeline creates spatial normalization parameters that are used to deform the TPMs, and the deformed TPMs are then used to guide the image deformation (Fig. 13) and the tissue segmentation (Fig. 14). In this experiment, to preserve the maximum tissue structures in the normalized and wrapped images, the bounding box of the volume is set to $(-90, -129, -110) \text{ mm} \times (90, 90, 98) \text{ mm}$ with voxel size $(1.0, 1.0, 1.0) \text{ mm}$.

5.2.4. Smoothing

To suppress noise in functional and anatomical images, a Gaussian kernel with full-width at half maximum (FWHM) $[4, 4, 4] \text{ mm}$ in x, y and z directions is used to smooth the spatially normalized and wrapped time-series BOLD fMRI images. The results are shown in the second row of Fig. 13.

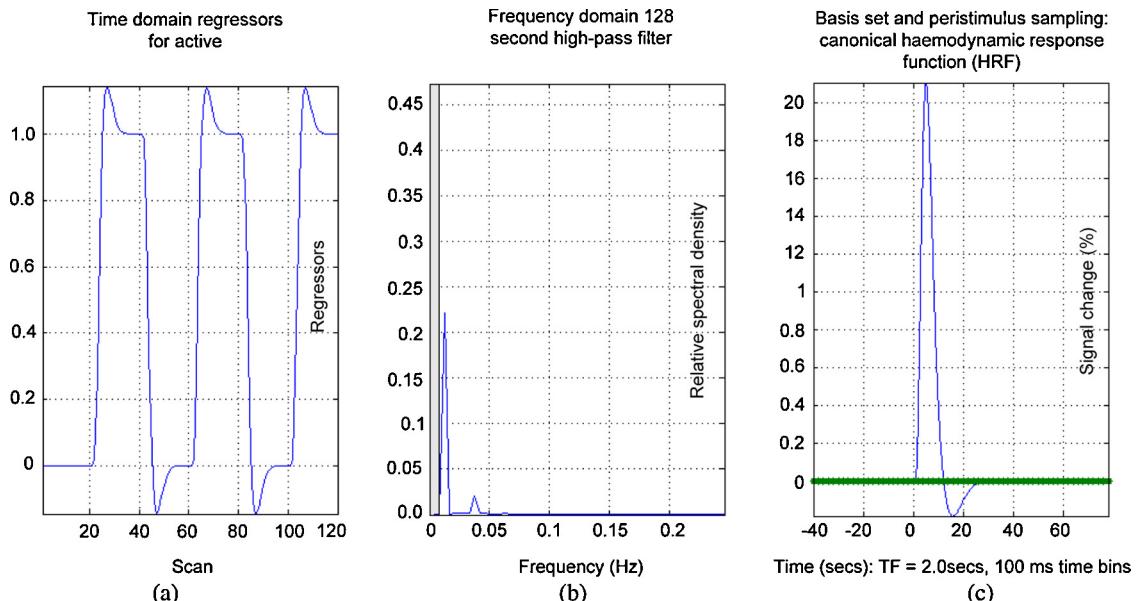


Fig. 15. The exploration results of the time-series BOLD fMRI image statistical analysis: (a) diagram describing the time series of the “active” regressor; (b) frequency domain plot of the active regressor; and (c) the basis function used to convert assumed neuronal activity into hemodynamic activity.

5.3. Statistical analysis

In this experiment, to detect task-related neural activation, the General Linear Model (GLM) [40] is used to statistically analyze the processed time-series BOLD fMRI images. The timing parameters used to construct the GLM design matrix: the design unit is “scan”, the interscan interval is 2 s, and the value used to set the microtime resolution and onset are 16 and 1 respectively.

Total 120 smoothed, normalized fMRI images are collected, and three active-scans (motor task) and three inactive-scans (static) are performed. The active-scan starting points (in slice) are 20, 60, 100 respectively with duration 20 slices, i.e., 40 s, in each of them. The canonical hemodynamic response function (HRF) is used as a basis function to model the hemodynamic response (Fig. 15(a)). The High Pass Filter (HPF) with 128 s, i.e., 0.008 Hz, is employed to remove the frequency content of the “active” regressor that is above the preset frequency (gray part in Fig. 15(b)), and Fig. 15(c) shows a frequency domain plot of the “active” regressor.

The analysis results are estimated with SPM. A t-contrast with $p\text{-value} = 0.05$ is used to control the Family Wise Error (FWE). To predict the neural signal response to foot and hand motor task, a “boxcar” function is employed in the model. Due to the GLM is normally used in a univariate way with the same correlation structure for every voxel, the estimation model only considers one voxel and therefore the fitting to the model is set to a single voxel’s time-course. Fig. 16 illustrates the fitted voxel responses across time with errors added (red points). Finally, the extracted time-series BOLD fMRI images of activation are superimposed on the registered, spatially normalized, and bias-corrected anatomical image with the results shown in Fig. 17.

6. Experimental analysis and results

As shown in Fig. 6, the entire anatomical MR image of brain, segmented grey matter, blood vessel, and two fMRIs are registered and displayed in real time, and enhanced with different color and opacity. The average multi-volume rendering speed is higher than 60 frames per second using an Nvidia GXT 480 graphics processing unit (GPU) with graphics driver version GeForce 285.62, delivering real-time performance.

To validate the new techniques developed in this software platform, we first visually compared the fusion and rendering results of multiple 2D slice images generated with different lookup tables. Next, we asked ten users to use our software platform to interactively visualize and explore the multi-function neurological images, and two optical mapping methodologies were employed in this process, i.e., the first is the traditional 2D and 3D independent (indep.) optical mapping approach, while the second is our new synchronized (sync.) optical mapping scheme. We compared the users’ operation performance and analyzed the results.

6.1. Lookup table and slice image fusion

We first compared the visualization results of the neurological images, whose color and opacity mapping was based on lookup tables generated with different algorithms. A total of four datasets were used in the experiment. The first layer, i.e., the lowest layer, is anatomical brain image (gray with opacity 1.0), the second layer is grey matter (yellow with opacity 0.25), the third layer is fMRI of foot motor task (green with opacities 0.6, 0.4, 0.2 and 0.8 for generating the fused images Fig. 18(a)–(d) respectively), and the fourth layer, i.e., the topmost layer, is the fMRI of finger motor task (magenta with the same opacities as those of the fMRI foot). Images in Fig. 18(a)–(c) are generated with the traditional VTK-based lookup

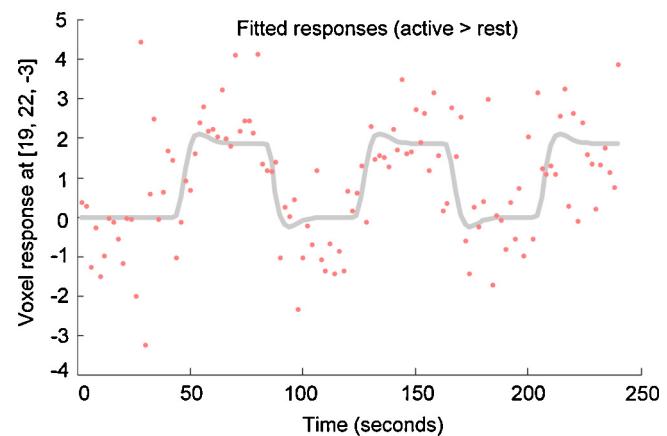


Fig. 16. Diagram illustrating the fitted response of a typical voxel in the time-series BOLD fMRI image statistical analysis process will errors added (red points) to the fitted response (gray line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

table as described in Section 4.2, while Fig. 18(d) is generated with the lookup table created with our opacity-adjustment algorithm.

We can see that when using VTK-based lookup table for color and opacity mapping, the fused image is dark and underlayer images are usually unclear, such as images Fig. 18(a) and (b). Decreasing the opacities of the top two layers will improve the dark image problem, such as changing the opacity from 0.6 to 0.2. However, the overlayed images with low opacity, such as 0.2, will become too light to be observed clearly, as illustrated in Fig. 18(c). At the same time, when using our opacity-adjustment algorithm, since we map the opacity of the black background color to zero, so the final fused image is clear with normal brightness even though when setting the opacities to be 0.8 in the top two layer images, as shown in Fig. 18(d).

6.2. Optical mapping and synchronization

We compared and analyzed the user perform variances of using different optical mapping and adjustment methodologies to visualize, manipulate, and explore multi-function neurological images. Ten medical imaging experts with knowledge of brain anatomy and function participated in the experiment. The users first loaded the main MR brain image along with the data sets of angiographic image, segmented white matter, and fMRI images of foot and finger motor tasks. Next, they adjusted the loaded data sets for image visualization and exploration using the same volume rendering algorithm and two different optical mapping methods.

The first method, i.e., IndepAlg, is commonly used in the existing software packages, i.e., the users manipulated the TFs to adjust and classify the loaded 3D neurologic data sets for volume rendering and data fusion. After finishing the adjustment of each of the loaded medical volumes, the users did the corresponding color and opacity mapping manipulation through adjusting 2D lookup table for slice image display, trying to achieve an identical 2D image fusion result as shown in their corresponding volume rendered images. In the second method, i.e., SyncAlg, the users employed our newly developed synchronized 2D and 3D optical mapping and adjustment scheme to explore the same data set. During the color and opacity classification and mapping process, the optical parameters used for volume rendering each of the above described 3D medical data sets were automatically used to build 2D lookup table for displaying and updating the corresponding slice images in real time.

Both of the 2D and 3D optical mapping algorithms are real time, no detectable time delay due to the computational cost of the color

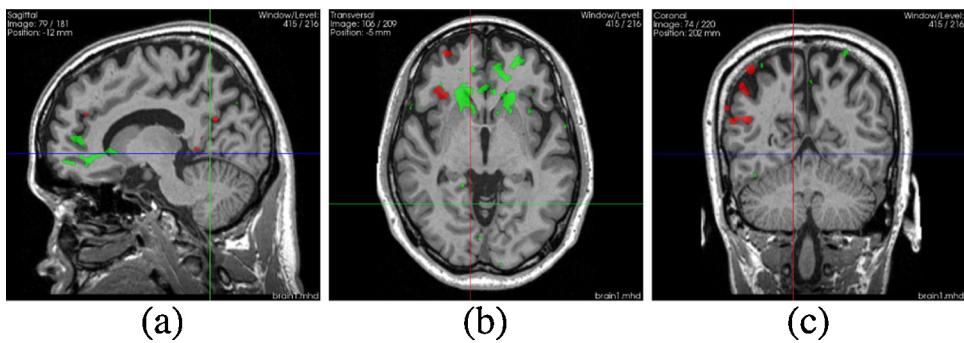


Fig. 17. The results of superimposing the extracted time-series BOLD fMRI images describing the canonical hemodynamic response of the motor task of foot (green) and finger (red) upon the spatially normalized and bias-corrected anatomical brain image of the same subject. The slice orientation: (a) sagittal, (b) axial (transverse), and (c) coronal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

and opacity classification. The operation time difference is mainly due to users' different manipulation skills and the application of our new 2D/3D optical mapping synchronization scheme. In addition, the order of method one IndepAlg and two SyncAlg are random, five users used IndepAlg first, and the other five employed SyncAlg first, so their performance improvement cannot be attributed to their training difference due to using one method after the other one.

Fig. 19 shows a snapshot of the optical mapping and volume visualization of the multi-function neurological images. The anatomical brain image is displayed and fused with segmented white matters, extracted fMRI images of foot and finger motor tasks, as well as blood vessels. Different colors and opacities are used to visualize the medical volumes and slice images, i.e., the main anatomical brain image is gray, white matters are brown, fMRIs of foot and finger motor tasks are green and magenta respectively, and blood vessel is red. The opacities of the fused images are set

to be 0.8. Fig. 19(a) illustrates the result of the multi-volume rendered neurological images generated through adjusting the color and opacity TFs for each of these fused images, and all of these optical parameters are automatically used to build the corresponding 2D lookup table for the display and fusion of the slice images at the locations of the three cross-volume planes as shown in Fig. 19(b)–(d) respectively.

When comparing these two optical mapping approaches, all the ten users expressed that when using the synchronized 2D/3D mapping scheme, they could concentrate on the data exploration and analysis instead of spending most of their time on software parameter adjustment. The left of the Table 1 shows the quantitative result of user performance, from which we can see that when using the traditional optical mapping method, it took users from ~20 s to ~49 s to adjust and display one neurological image in both 2D and 3D rendering windows. When they used our new synchronized optical mapping method, it generally took the users 50% less time

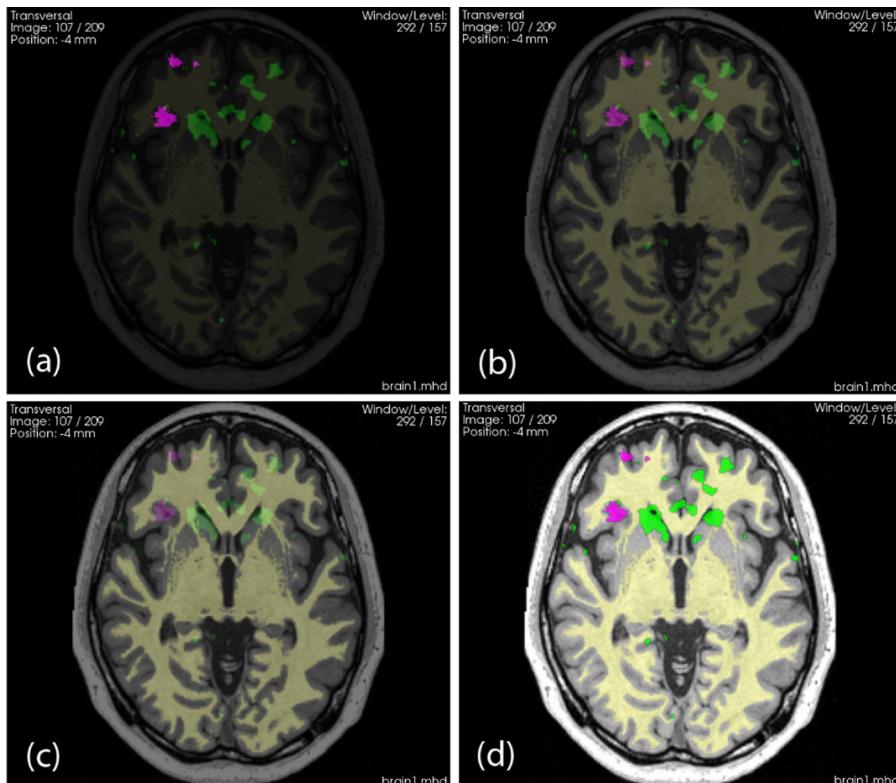


Fig. 18. The visual comparison of multiple slice image fusion results. The final images are created with lookup tables that are built with two different algorithms.

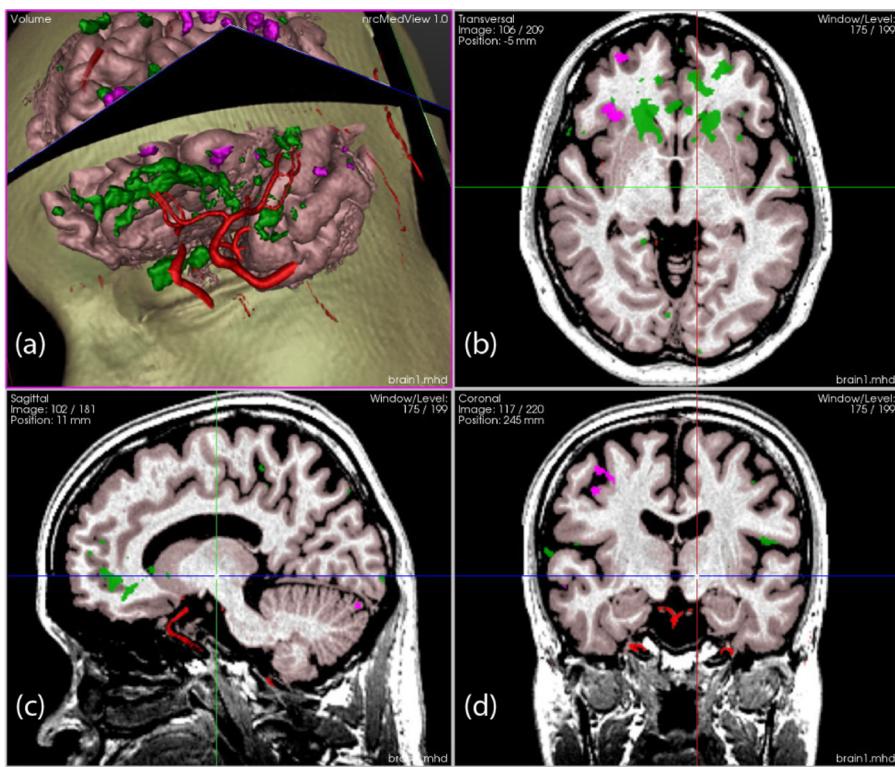


Fig. 19. The snapshot of the multi-function visualization and fusion result of both 3D rendering (a) and three 2D slice image display (b)–(d), which are generated with the synchronized optical mapping algorithm.

Table 1

The individual user operating time (in second) and the mean with standard deviation (SD) of the average user performance in optical mapping and adjustment for the four neurological images using the traditional 2D/3D independent (indep.) and our new synchronized (sync.) optical mapping algorithms (left and middle of the table), as well as the corresponding mean operating time and SD ratio percentage of using our new synchronized (sy. or sync. in second) 2D/3D optical mapping scheme vs. using the traditional independent (in. or indep. in second) optical mapping method (right of the table).

Image type Neurological image names	Optical mapping and adjustment														Percentage (%)											
	User 1		User 2		User 3		User 4		User 5		User 6		User 7		User 8		User 9		User 10		Indep.	Sync.	Sync. vs. Indep.			
	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	In.	Sy.	Mean	SD	Mean	SD	Mean	SD		
Grey matters	25	11	32	14	28	10	34	18	20	12	30	18	51	16	27	12	25	12	28	13	30.0	8.35	13.6	2.84	45.33	33.33
fMRI foot	30	15	36	22	35	15	37	20	21	12	31	16	56	30	23	13	21	12	31	15	32.1	10.3	17.0	5.60	52.96	54.37
fMRI finger	25	13	28	14	33	14	26	19	24	14	25	18	40	26	29	15	23	12	31	15	28.4	5.17	16.0	4.11	56.34	78.85
Blood vessel	34	13	32	13	43	15	49	16	36	12	34	14	53	23	44	12	34	12	38	15	39.7	7.20	14.5	3.31	36.52	45.83

to achieve the same visual results, even though the actual time consumed of each user depends on his/her manipulation skills and the data used in the experiment.

Fig. 20 and the middle and right of **Table 1** show the user's average time with the standard deviation (SD) that were used to adjust the 3D and 2D optical mapping for neurological image display and capturing the structure of interest. The 2D/3D independent optical mapping algorithm, i.e., IndepAlg, and our 2D/3D synchronized optical mapping scheme, i.e., SyncAlg, were used in the experiment. When using the IndepAlg, we note that the users cost similar time to adjust the images of grey matter, fMRI foot and finger motor tasks, i.e., using ~30 s with SD 1.86. This is due to the fact that the users' operations are similar in adjusting and exploring these three images. However, for blood vessel display, it took user ~32% more time, we attribute this to the fact that the voxel intensity of interest of the angiography data locates at a narrower range than those of the other three images, so the normal window-level settings cannot be directly used to visualize the data correctly. The users therefore needed to first adjust the window-level to capture the intensity range of interest, and then they had to perform the

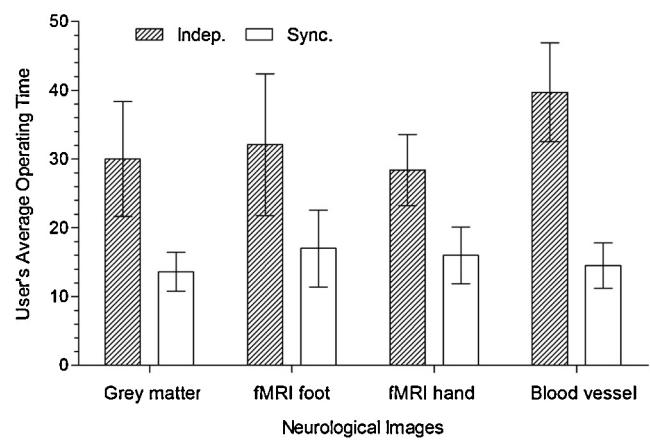


Fig. 20. The comparison of the user's average operating time (in second) with standard deviation (SD) for adjusting the 2D/3D optical mapping for multi-function neurological image fusion and visualization through using the traditional independent (in. or indep.) 2D and 3D optical mapping algorithm and our new synchronized (sy. or sync.) optical mapping scheme.

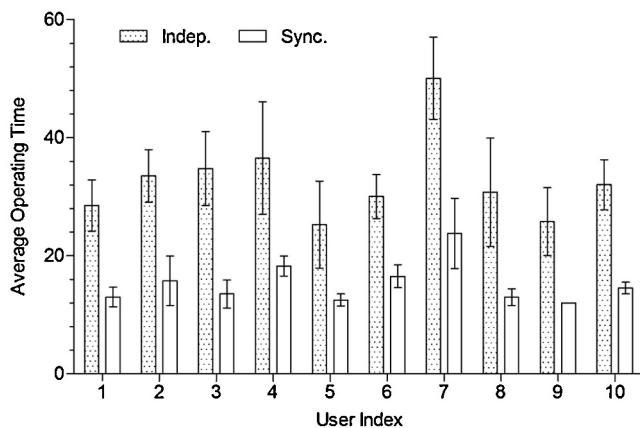


Fig. 21. The comparison of user's average time (in second) with standard deviation (SD) for adjusting the four neurological data sets using the traditional independent (In. or Indep.) 2D and 3D optical mapping methods and our new synchronized (Sync. or Sync.) 2D/3D optical mapping scheme.

similar operations as they did in the manipulation of the other three images.

For each of the four neurological images, when compared with the users' performance of using the IndepAlg, we note that their performance is improved with the use of the SyncAlg, i.e., the user's average operating time decreases from 28.4–39.4 s to 13.6–17 s. At the same time, we also note that the operating time SD range narrows from 5.17–10.3 to 2.84–5.6, i.e., the SDs are 33.33–78.85% of their corresponding values of using the IndepAlg, which means that the user's operating performance is more uniform when using the new SyncAlg than that of using the traditional IndepAlg. Furthermore, when using the SyncAlg, the highest improvement is blood vessel extraction, using only 36.54% of the time of using the IndepAlg. This is due to the fact that when using the SyncAlg, the users did not need to adjust the window-level, and we also implemented a pre-set color and opacity classification scheme in the software platform, i.e., we adjusted the color and opacity mapping for the angiographic image volume rendering and slice image display beforehand, and saved the adjustment parameters into a XML file. During the experiment, the users loaded these pre-saved parameters directly through opening the saved XML file without additional operations.

Fig. 21 illustrates each of the ten users' mean operating time for adjusting the four neurological images, including the use of the traditional 2D/3D independent optical mapping algorithm IndepAlg and the new synchronized optical mapping scheme SyncAlg. Table 2 shows the quantitative analysis results. For all these ten users, we can observe their performance improvement when using the new SyncAlg to adjust the four neurological images. When using the IndepAlg, the user's mean operation time is from 25.25 s to 50 s; while using the new scheme, their mean operation time is from 12 s to 23.75 s. When compared with the operating time of using the IndepAlg, the users need only ~45% of their original operating time to finish the same operations through using the new mapping algorithm SyncAlg. Furthermore, the user's operating time SD of using the new synchronized mapping scheme SyncAlg is narrower than that of using the traditional independent optical mapping method IndepAlg. When using the IndepAlg, the user's average mean operating time SD and the SD of the error (SDE) are 7.08 and 2.24 respectively, while using the new SyncAlg, their corresponding average mean operating time SD and SDE are 3.57 and 1.13. The SD and SDE decrease rates are 49.58% and 49.55% respectively.

In addition, in Table 2, we also calculated the paired *t*-test of the users' mean operating time of manipulating the 4 neurological image datasets used in our experiment, getting *p*-value ~1.7e-06,

Table 2

The mean time (in second) with standard deviation (SD) of each of the ten users' average manipulation time in adjusting the four neurological images, along with the operating time and SD ratio percentage of using our new synchronized (sync.) optical mapping scheme vs. using the traditional independent (indep.) 2D/3D optical mapping method.

User index	Optical mapping and adjustment				Percentage (%)	
	Indep. (s)		Sync. (s)		Sync. vs. Indep.	
	Mean	SD	Mean	SD	Mean	SD
1	28.50	4.36	13.00	1.63	45.61	37.39
2	33.50	4.44	15.75	4.19	47.01	94.37
3	34.75	6.24	13.50	2.38	38.85	38.14
4	36.50	9.54	18.25	1.71	50.00	17.92
5	25.25	7.37	12.50	1.00	49.50	13.57
6	30.00	3.74	16.50	1.92	55.00	51.34
7	50.00	6.98	23.75	5.91	47.50	84.67
8	30.75	9.18	13.00	1.41	42.28	15.36
9	25.75	5.74	12.00	0.00	46.60	0.00
10	32.00	4.24	14.50	1.00	45.31	23.58

which is much smaller than the predetermined significance level of 0.05 or 0.01, so we can reject the null hypothesis, i.e., when compared with the traditional independent 2D/3D optical mapping algorithm, our new synchronized technique can improve the user's performance in exploring multimodality neurological images. Furthermore, the calculated 95% confidence interval is ~[12, 22]. It covers most of the experimental data range, which indicates that the experimental estimation has high reliability and there is statistically significant user performance improvement when using our new synchronized 2D/3D optical mapping method.

We therefore conclude that when using the new SyncAlg, the user's performance can be improved in both the operating time's length and uniformity, at the same time, they can achieve the same or even better color and opacity mapping and 2D/3D visualization results in the final fused multi-function neurological images.

7. Conclusions

In this paper, we described the development and performance of a comprehensive software platform for the visualization and exploration of multi-function neurological data sets. The software was designed with an object-oriented modular structure that was transparent and extensible to new functions and working pipelines. In the visualization process, we incorporated our artifact-free interactive voxel classification algorithm into a GPU-based raycasting for real-time volume rendering, delivering high-quality images.

When multiple 2D medical images are overlaid, the resultant fusion image usually suffers from a zero-opacity problem, i.e., the underlayer images cannot be viewed clearly, we therefore presented an opacity-adjustment algorithm for building lookup tables with new optical mapping rules to solve this problem, resulting in improved multi-slice image display without blurring the underlayer images or darkening the fusion results. To address the issues such as inefficiency, classification errors, and inconsistency in the multi-function medical image color and opacity mapping and visualization, we proposed an algorithm to synchronize the 3D voxel optical classification for volume rendering with the building of 2D color and opacity lookup table for slice image display, so the optical mapping in both 3D volumes and 2D slice images can be updated simultaneously in real time using the same adjustment operations. The users therefore can visualize and explore multiple images in both 3D and 2D rendering windows at the same time with improved fidelity in the resulting fusion images. Furthermore, our software works seamlessly with the medical data processing and statistical analysis pipeline, providing an integrated framework for multi-function neurological image processing and exploration,

allowing neurosurgeons to flexibly, intuitively, and rapidly view, manipulate, and analyze functional and structural MR images.

The developed software platform with new tools for medical image visualization, fusion, synchronization, and exploration will provide clinicians with a deep insight into the medical data sets, improve their understanding of the relationships between brain structures and functions, and enhance their ability of precisely locating lesions, knowing their spatial relationships with healthy tissues, and effectively identifying eloquent cortex in close proximity to the intracranial lesions.

Conflict of interest

There is no conflict of interest.

Acknowledgments

The authors would like to thank Dr. Sboto-Frankenstein's clinical validation of the fMRI analysis results, and will also thank our colleagues at the National Research Council Institute for Biagnostics, Winnipeg, Canada, for validating and testing our developed algorithms and software platform.

References

- [1] Cleary K, Peters TM. Image-guided interventions: technology review and clinical applications. *Ann Rev Biomed Eng* 2010;12:119–42.
- [2] Dimaio S, Archip N, Hata N, Talos I-F, Warfield S, Majumdar A, et al. Image-guided neurosurgery at brigham and women's hospital. *Eng Med Biol Mag IEEE* 2006;25:67–73.
- [3] The use of neuroimaging to study behavior in patients with epilepsy. *Epilepsy Behav* 2008;12:600–11.
- [4] Logothetis NK. The underpinnings of the BOLD functional magnetic resonance imaging signal. *J Neurosci* 2003;23:3963–71.
- [5] Pujol S, Kikinis R, Gollub R. Lowering the barriers inherent in translating advances in neuroimage analysis to clinical research applications. *Acad Radiol* 2008;15:114–8.
- [6] Enchev Y. Neuronavigation: geneology, reality, and prospects. *Neurosurg FOCUS* 2009;27:E11.
- [7] Rorden C, Brett M. Stereotoxic display of brain lesions. *Behav Neurol* 2000;12:191–200.
- [8] Moreland K. The paraview tutoria, Version 3. 10. Sandia National Laboratories; 2010 <http://www.vtk.org/Wiki/images/2/2d/ParaViewTutorial310.pdf>
- [9] Visualization Toolkit (VTK) 5.8.0. Clifton Park, NY, USA: Kitware Inc.; 2011 <http://www.vtk.org>
- [10] 3DSlicer Version 3.6. Brigham and Women's Hospital and 3D Slicer Contributors; 2011 <http://www.slicer.org/>
- [11] Gouws AD, Woods W, Millman RE, Morland AB, Green GGR. Dataviewer3d: an open-source, cross-platform multi-modal neuroimaging data visualization tool. *Front Neuroinformatics* 2009;3.
- [12] Amira, 2011. Visage Imaging, GmbH <http://www.amira.com>
- [13] Mahmoudi SE, Akhondi-Asl A, Rahmani R, Faghhi-Roohi S, Taimouri V, Sabouri A, et al. Web-based interactive 2d/3d medical image processing and visualization software. *Comput Methods Prog Biomed* 2010;98:172–82.
- [14] Insight segmentation and registration toolkit (ITK) 3.20. The National Library of Medicine; 2010 <http://www.itk.org>
- [15] BrainVoyager QX. Netherlands: Brain Innovation B.V.; 2011 <http://www.brainvoyager.com>
- [16] Friston K, Ashburner J, Kiebel S, Nichols T, Penny W, editors. *Statistical parametric mapping: the analysis of functional brain images*. Academic Press; 2007.
- [17] Pyka M, Hertog M, Fernandez R, Hauke S, Heider D, Dannlowski U, et al. fMRI data visualization with brainblend and blender. *Neuroinformatics* 2010;8:21–31, <http://dx.doi.org/10.1007/s12021-009-9060-3>.
- [18] Hess R. *The essential blender: guide to 3D creation with the open source suite blender*. No Starch Press; 2007.
- [19] Woolrich MW, Jbabdi S, Patenaude B, Chappell M, Makni S, Behrens T, et al. Bayesian analysis of neuroimaging data in FSL. *NeuroImage* 2009;45:173–86. *Mathematics in Brain Imaging*.
- [20] FSLView v3.1. Netherlands: Brain Innovation B.V.; 2010 <http://www.fmrib.ox.ac.uk/fsl/fslview>
- [21] Analysis of Functional NeuroImages (AFNI); 2011 <http://www.afni.nimh.nih.gov/afni>
- [22] Rößler F, Tejada E, Fangmeier T, Ertl T, Knauff M. Gpu-based multi-volume rendering for the visualization of functional brain images. In: Schulze T, Horton G, Preim B, Schlechtweg S, editors. *SimVis*. SCS Publishing House e.V.; 2006. p. 305–18.
- [23] Beyer J, Hadwiger M, Wolfsberger S, Bühl K. High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. *IEEE Trans Vis Comput Graph* 2007;13:1696–703.
- [24] Köhn A, Weiler F, Klein J, Konrad O, Hahn HK, Peitgen H-O. State-of-the-art computer graphics in neurosurgical planning and risk assessment. In: Cignoni P, Sochor J (editors). *Eurographics short papers and medical prize awards*. p. 117–20.
- [25] Rieder C, Ritter F, Raspe M, Peitgen H-O. Interactive visualization of multimodal volume data for neurosurgical tumor treatment. *Comput Graph Forum* 2008;27:1055–62.
- [26] Rieder C, Schwier M, Hahn HK, Peitgen H-O. High-quality multimodal volume visualization of intracerebral pathological tissue. In: Botha CP, Kindlmann GL, Niessen WJ, Preim B, editors. *Visual Computing for Biology and Medicine (VCBM)*. Eurographics Association; 2008. p. 167–76.
- [27] Hergelegiu GMMPV. Visualization of segmented structures in 3D multimodal medical data sets. *Adv Electr Comput Eng* 2011;11:99–104.
- [28] Diepenbrock S, Prašný J-S, Lindemann F, Bothe H-W, Ropinski T. Interactive visualization techniques for neurosurgery planning. UK: Eurographics Association, Llandudno; 2011. p. 13–6.
- [29] Alper Selver M, Fischer F, Kuntalp M, Hillen W. A software tool for interactive generation, representation, and systematic storage of transfer functions for 3d medical images. *Comput Methods Prog Biomed* 2007;86: 270–80.
- [30] Pinto MFd, Freitas CMDS. Volume visualization and exploration through flexible transfer function design. *Comput Graph* 2008;32:540–9.
- [31] Zhang Q, Eagleson R, Peters TM. Rapid scalar value classification and volume clipping for interactive 3D medical image visualization. *Vis Comput* 2011;27:3–19.
- [32] Zhang Q, Eagleson R, Peters TM. Dynamic real-time 4D cardiac mdct image display using GPU-accelerated volume rendering. *Comp Med Imag Graph* 2009;33:461–76.
- [33] OpenGL 4.2. USA: Silicon Graphics International Corp.; 2011 <http://www.opengl.org>
- [34] OpenGL Shading Language (GLSL). USA: Silicon Graphics International Corp.; 2011 http://www.opengl.org/wiki/OpenGl_Shading_Language
- [35] Qt 4.7.4. Finland: Nokia Corporation; 2011 <http://www.qt.nokia.com>
- [36] The Common Toolkit (CTK); 2011 <http://www.commonkt.org>
- [37] The Medical Imaging Interaction Toolkit (MITK), 0.14. Heidelberg, Germany: Division of Medical and Biological Informatics German Cancer Research Center; 2009 <http://www.mitk.org>
- [38] Zhang Q, Eagleson R, Peters TM. Rapid scalar value classification and volume clipping for interactive 3d medical image visualization. *Vis Comput* 2011;27:3–19.
- [39] Rorden C. dcm2nii DICOM to NIfTI conversion. In: The source for neuroimaging: MRICron; 2009.
- [40] Worsley KJ, Friston K. Analysis of fmri time-series revisited – again. *NeuroImage* 1997;2:173–81.