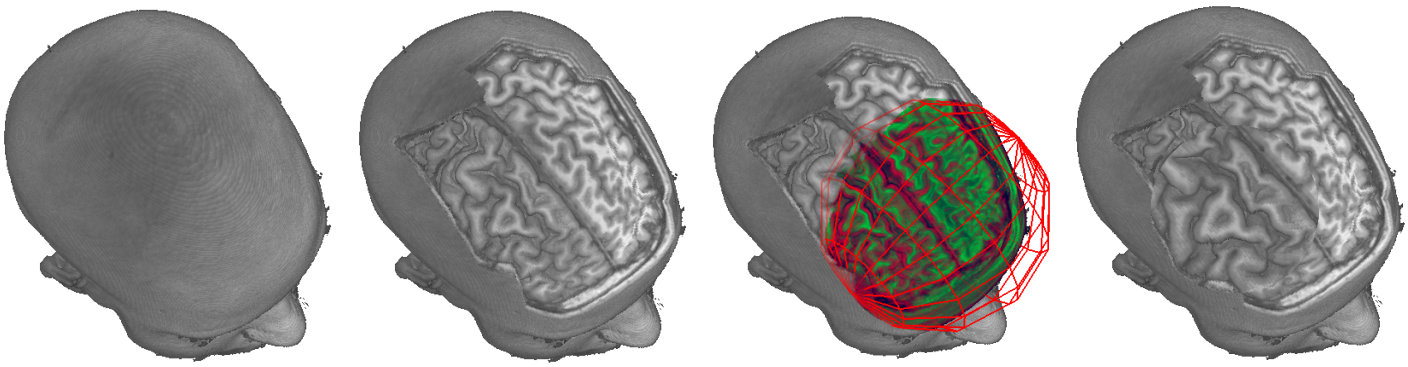
Query Tools for Interactive Exploration of 3D Neuroimages: Cropping, Probe and Lens

Wu, Shin-Ting*, José Elías Yauri Vidalón*, Wallace Souza Loos and Ana Carolina Coan†
*School of Electrical and Computer Engineering, University of Campinas, Campinas, Brazil
†School of Medical Sciences, University of Campinas, Campinas, Brazil



(a) (b) (c) (d)

Fig. 1. Interactions on (a) 3D neuroimage: (b) cropping to reveal hidden region, (c) a probe to highlight an area of interest, and (d) a lens to magnify details.

***Abstract*—Dynamic queries continuously update the data that is visualized in accordance with the user actions. They are typically applied for visual information seeking. This paper proposes to introduce this interaction style for exploring 3D medical neuroimages in its original form, enhancing visual seeking technology in a medical diagnostic procedure. More precisely, we present three dynamic query tools that allow the user to change the focus on-the-fly, while the surrounding tissue is preserved. They are a curvilinear cropper, a volumetric probe and a movable magnifying lens. Once information-preserving visualization is essential for accurate diagnosis and legal protection, a dataset is in its original form. The originality of our work relies on the input interface through which an expert can directly manipulate those tools on the raw data and the responsiveness of each displayed voxel by exploiting the power of GPUs. The proposed techniques have been integrated in a visualization prototype and were assessed by the neuroimaging experts, who were be able to identify subtle lesions in the brain.**

***Keywords*-Focus+context techniques, Probe, Lens, Curvilinear cropping, Computer-aided diagnosis, Magnetic resonance neuroimages.**

I. Introduction

Because of its high spatial and spectral resolution, it is increasing the use of magnetic resonance images (MRI) both in the study of human organs as well as in the diagnosis of structural and functional abnormalities. Along with the rapid evolution of medical image processing algorithms, computer-aided diagnostics systems specialized in mammography and angiography have emerged over the past years. However, the structural complexity of the brain, a great variety of brain lesions and individual anatomical shape of skulls are major challenges for developing a diagnostic system specializing in neuroimages. Expert interventions are still essential both in the identification and in the interpretation of neurological findings. And, for accurate diagnosis and legal protection, the clinicians are particularly interested in having an alternative to view volume data in its original form. Therefore, not only the visual effects, but also the way that a user can obtain the desired visual effects by controlling a 2D cursor pointer with a mouse plays an important role in the design of a user-friendly interface. In this paper we consider dynamic queries adequate responses of a raw volume dataset while a user hovers the 2D cursor pointer over its viewable voxels, such as in [1].

Dynamic queries which enable physicians to gain insight into complex internal structure in its original form is actually at its very beginning stages. Focus+context visualization and volume clipping are representative efforts aiming at these techniques. Multiple rendering modes [2], [3], multiple magic volume lenses [4], [5], and multiple transfer functions [2] have been proposed to emphasize important parts of a volume (*focus*), whereas the reminder (*context*) is deemphasized. Also, several clipping algorithms have been developed to selectively remove parts of volume data for revealing and exploring hidden regions, such as confocal volume rendering [6], multi-planar reformation [7], and clipping in pre-specified arbitrary geometry [8], [9]. Exploded views for volume data have been presented as a solution to occlusion problem, too [10]. To our best knowledge, concerning the user's input style least amount

of attention has been given so far.

Our main concern is appropriate 3D interaction techniques that allow a neuroscientist, with the help of ubiquitous 2D devices, to quickly scan in native space a patient's original brain dataset. The key feature of our tools is that the user may manipulate the displayed object as if it were in her/is hands even when the underlying data is the raw data, not the data that is actually viewed. Without prior segmentation, a medical expert may, just by hovering the 2D cursor pointer over the data of interest, clean out the noises, make skin incision, remove the delineated head's bone, highlight regions of interest, and magnify the object for closer inspection.

In particular, we present in this paper one improved cropping tool, as shown in Fig. 1.(a), and two novel dynamic query tools to explore 3D medical volume datasets: one for highlighting features of interest (Fig. 1.(c)) and another for magnifying regions over which the cursor pointer is hovered (Fig. 1.(d)). Differently from the previous works, both dynamic query tools can slide smoothly over viewed 3D data according to mouse movement in the patient's space. Combining them with an appropriate cropping algorithm makes easier not only the finding of a subtle cortical lesion and the measurement of its spatial extent, but also the assessment of its depth as well.

Our algorithms are based on a ray casting paradigm in which the same ray traversal procedure is applied independently on each viewing ray shot from every pixel. Since the single instruction, multiple data (SIMD) architecture of GPU is a perfect fit for parallel numerical processing of such a massive dataset, we propose a GPU-based implementation with extensive use of non-displayable frame buffer object.

Contributions: Aiming at aiding the physicians and surgeons to explore, analyze and diagnose brain lesions from 3D MR neuroimages, the main contribution of this paper is a GPU-based implementation of two dynamic query tools, namely the *volumetric probe* and the *movable magnifying lens*, that guarantees befitting interactive responsiveness while a user displaces the 2D cursor pointer on displayed voxels. The 2D cursor is controlled with ubiquitous 2D pointing devices, such as a conventional mouse, making the direct manipulation simpler to learn. In order to make the proposed tools effective for investigating the surface of cerebral cortex, we also improve the curvilinear reformatting algorithm proposed in [1].

A. Related work

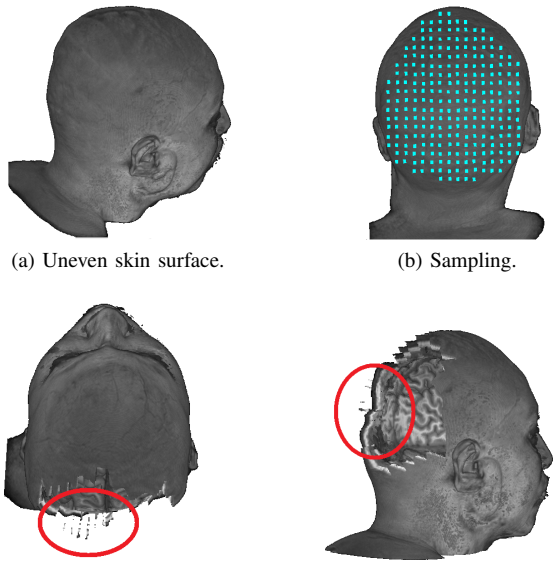
Zhou *et al.* adopt geometry-based approach to divide the volume data into focal and context regions, and render them with direct volume rendering and non-photorealistic techniques, respectively [3]. For highlighting the focal region, they use a distance-based opacity modulation from the center of geometry over the homogeneous region of the illustrated context. Viola *et al.* introduce the concept of importance-driven volume rendering in which 3D importance function is applied to assign visibility priorities, which control the sparseness of the voxel display along each viewing ray [11]. They assume the regions of interest within a volume are pre-segmented, and a user should attribute an importance value to each region. Bruckner and Gröller present a dynamic 3D illustration environment and exploded views system, which operate on pre-segmented volume data [12]. Recently, Sikachev *et al.* introduce a dynamic focus+context approach that highlights salient features during user interaction [13]. Nevertheless, the interaction is limited to affine transformations of the proxy geometry and not of the geometry that is actually displayed. In this paper, we present a volumetric probe with which a user can cause the focus to change as the cursor is dragged by a ubiquitous mouse.

Due to the growth in size and resolution of the volume datasets, another focus+context visualization style is obtained with use of magic lenses. They allow the user to magnify features of interest, without suppressing the remaining volume data. LaMar *et al.* present a hardware-texture based volume lens [4], while Wang *et al.* extend the idea to various standard and advanced magnification lens and integrate them into a GPU-based volume rendering algorithm [5]. Cohen *et al.* conducted an interesting comparative study of different lens effects on members of a neurosurgery team [14]. The focus of the mentioned works is, however, on visualization. In this paper we are interesting on the other half of interactive visualization: which actions a user should take to get the expected visual results. We propose a movable magnifying lens that may be applied on curvilinearly cropped regions presented in [1] to provide views that the medical doctors would have in the operating room. It lets the clinicians not only change the focus by simple drag as well as drill down in order to find data that may reveal any functional or structural abnormality.

As the user of our proposed tools is interacting directly with the raw data, and not the brain skin surface, the segmentation-free confocal volume rendering proposed by Mullick *et al.* does not fit to our needs [6]. Therefore, we applied the cropping algorithm presented in [1] to construct partially the brain skin surface from the data provided by a doctor in image space, as shown in Fig. 2.(b). For assigning correct depth value to each voxel with respect to the input samples, a mesh is constructed to tackle the discrepancy between the human perception and the computational representation. The built mesh serves as reference for generating a series of equally spaced offset meshes toward the head's midpoint. The mesh vertices are displaced in their normal direction, which is the average of the normal vectors of its adjacent triangular faces. Each offset mesh is then voxelized, and volume cells are labeled with the corresponding head's layer depth value. Fig. 2.(c) and (d) illustrate the removed regions in distinct views.

The described cropping procedure does its job quite well, except when the patient's scalp is uneven and presenting sharp edges, as depicted in Fig. 2.(a). Unwanted noises preventing thorough examination may result, as highlighted in Fig. 2.(c) and Fig. 2.(d). Kim *et al.* observe in their work that this is due to the fact that the vertices are moved along a single direction and propose using multiple normal vectors to offset a vertex [15]. An allowance parameter δ is defined to distinguish vertices with a single normal vector from the ones with

multiple vectors. To circumvent this ad-hoc classification, we use a displacement–decimation offset mesh algorithm.



(a) Uneven skin surface. (b) Sampling.

(c) Artifacts (top view). (d) Artifacts (side view).

Fig. 2. Curvilinear reformatting.

With graphics processing units (GPUs), it is possible to compute in real-time the projected voxel depth under the corresponding cursor. Performing 3D interactions with traditional 2D pointing devices becomes feasible. To interact more precisely with the visualized data, the idea of snapping a 3D cursor onto the visible fragment was introduced in [16]. Later, Wu *et al.* show how to generate a depth map of the viewed data at each user's interaction [17]. In this paper, we further extend these ideas to the development of 3D dynamic query tools.

B. Technique overview

As a setup for a brain focus+context visualization, we should transfer the 3D medical volume data and two transfer functions into GPU for ray-casting based volume rendering. One transfer function, called focal transfer function, is applied on the focal region and the other, denominated contextual transfer function, is used to map the data in the contextual region onto the optical properties consisting of color and opacity.

To enhance the image quality, the unwanted noises in the raw medical dataset may be filtered out by the user. And, to make the mapping more flexible, the transfer functions are editable interactively on CPU. In addition, a variety of depth maps are rendered and stored into an offscreen frame buffer object. These data are used to control the actions of the query tools manipulated by a user. The visual feedback of each valid user's input event is rendered and transfered to the onscreen frame buffer for displaying. The procedure is repeated until the user stops generating input events. When this repetition occurs at interative rate, the system causes the feeling that it promptly responds user's actions.

An overview of our proposed interaction architecture is shown in Fig. 3.

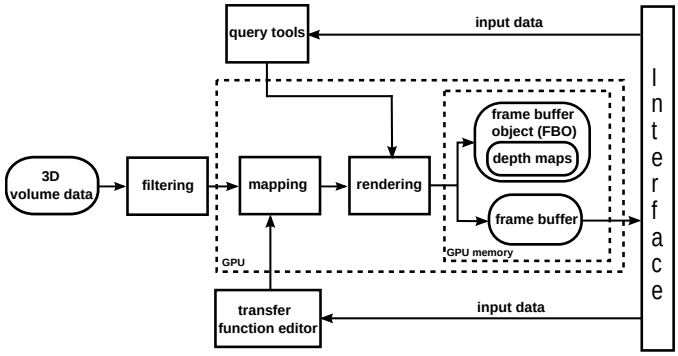


Fig. 3. Interaction architecture.

II. Technical Background

In this section, we summarize some important concepts that are necessary to understand our proposal and the specific medical requirements.

The basic goal of GPU-based volume rendering is to estimate per pixel the light intensity that reaches the viewer after traversing the volume data along the light ray in the object space. One pragmatic approach is to resample the volume data at regular intervals along the ray and compute the light's contribution of each traversed voxel. By means of an appropriate transfer function, color $C_{src} = (R_{src}, G_{src}, B_{src})$ and opacity α_{src} are assigned to each sample. In particular, when R_{src}=G_{src}=B_{src} we have a grayscale image, when α_{src}=1.0 the voxel is totally opaque, and when α_{src}=0.0 it is totally transparent. These optical properties are recurrently composed in the same order as the ray traversal order, usually in the front-to-back one, to provide a final intensity C_{dst} and opacity α_{dst} of each pixel [18].

It is worth remarking that the grayscale and non-composition rendering is preferred by the medical experts. Preference toward grayscale is because the physicians are trained to obtain findings on these kinds of images. And the composition mode, though providing an overview of 3D volume data, may blend out subtle brain abnormalities.

As shown in Section III-A, the user either brushes the volume data or specifies the position of the query tools on the 2D projected image by dragging a 2D cursor. The window manager (WM) on CPU is responsible for handling all these events. On one hand, using ubiquitous 2D devices demands less amount of effort for a neuroexpert to master the new exploration tools. On the other hand, it requires each 2D cursor position on a visible voxel to be mapped onto 3D position in the patient's native space. In our work, we apply the technique presented in [17], which recovers 3D coordinates (x, y, z) from the cursor screen position (u, v) and its corresponding $depth$ stored in a frame buffer object.

The frame buffer object (FBO) is a mechanism for rendering to one or more offscreen objects [19]. Offscreen rendering

simply means that the content of the frame buffers is not visible until it is brought back, switching to the default onscreen frame buffer. Similar to the onscreen frame buffer, a FBO contains a collection of rendering destinations: color, depth and stencil buffers. Of our great interest is its capability to perform offscreen rendering of the depth maps. It allows us to construct as many depth images as necessary and to store them as 2D textures or pixel arrays for further processing.

III. New techniques

In this section we detail our solutions for the proposed dynamic query tools. First of all, we present an overview of their interaction sequences. The key idea of our proposal is to aggregate to each displayed voxel its depth value through appropriate depth maps. In addition, we exploit the processing power of GPU for performing several intermediary numerical computations, which ensure an interactive responsiveness of each volume data sample.

A. Interaction Sequences

For revealing the hidden structures, the user must first define the region of interest (ROI) by brushing the cursor over the visualized patient's head surface. When the action is concluded, a series of layers parallel to the brushed region is constructed from the sampled pixels and voxelized in the resolution of the 3D volume data. The peeling depth is, then, assigned to each voxel and transfered to the GPU memory. Through a slider, the user can interactively select a crop depth value. This makes all voxels with tag less or equal to the selected value invisible. Hence, the outcome is a curvilinear cropped head. This action sequence, including the rendering setup, are summarized in the sequence diagram depicted in Fig. 4.

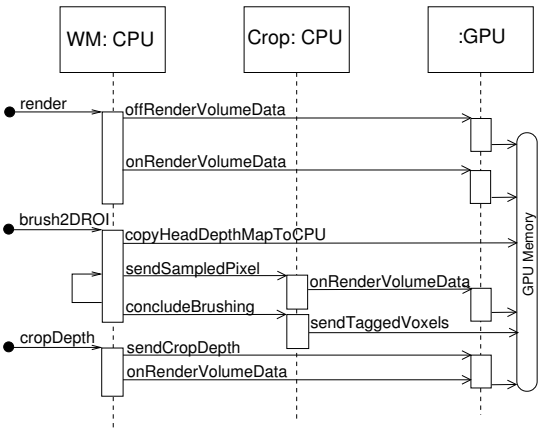


Fig. 4. Sequence diagram of cropping.

The user can further choose either *volumetric probe* or *movable magnifying lens* query mode for investigating brain internal structures. Their rendering is performed in two steps. First, the depth maps of the tool geometry are generated to help 3D movement control. Then, visual feedback to a user's action is rendered. Moreover, for focus+context visualization, the focal and the contextual transfer functions, represented as two look-up tables, are transfered to GPU.

If the volumetric probe is chosen, the user should further decide the probe geometry parameters. As visual feedback, the probe pops out, as illustrated in Fig. 1(c), and the user can drag it in any direction by just making continuous mouse movement. The intersection of the probe geometry and the 3D volume data is rendered with the focal transfer function, while the remainder with the contextual transfer function. As the probe geometry changes according to mouse movement, as shown in Fig. 5, the user has the perception that the focus is modified interactively under her/is control. Explanation for twice offscreen rendering the probe geometry is given in Section III-C.

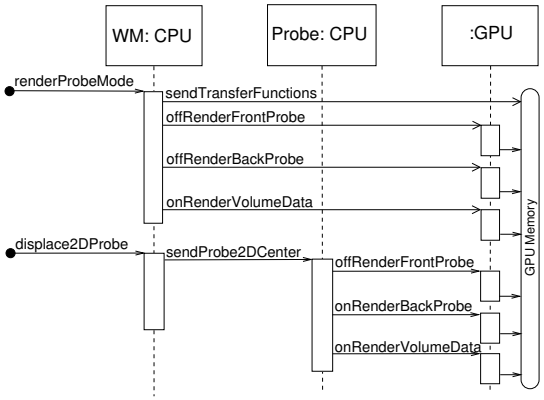


Fig. 5. Sequence diagram of probe displacement.

When the user decides for the movable magnifying lens, a disk pops out on the visible volume surface, as illustrated in Fig. 1(d). It slides over the surface in accordance with mouse movement. The intersection of this disk and the visible voxels is rendered with a perspective view, so that the user has the perception that the voxels covered by the lens are magnified. Moreover, non-composition is performed on the colors of the voxels inside the disk for preserving their original information. Fig. 6 sketches the CPU and GPU cooperation. We will see in Section III-C why the depth map of the volume data is copied back to CPU.

B. Formulation

In view of the limitation of human processing ability, focus+context visualization has been shown suitable to many situations [20]. For neuroimage based diagnosis, for example, it enables a neuroscientist to examine a region of interest in full detail, while maintaining an overview of the head for comparative analysis. However, four issues related with dynamic queries should be addressed. First of all, how can a doctor interactively peel the brain's layers without unwanted artifacts. Second, how can a physician interactively change his focus with simple mouse movement in the volume data domain. Third, how to preserve original information in the focal region for more accurate examination without loss of

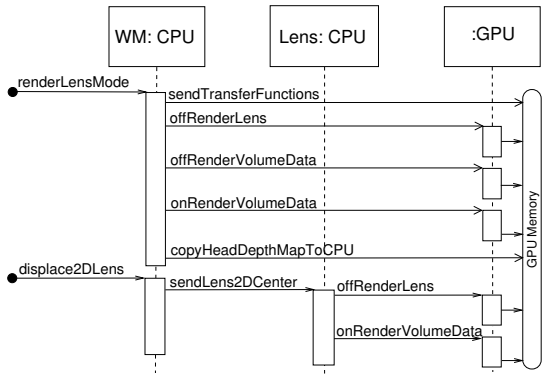


Fig. 6. Sequence diagram of lens displacement.

context. Fourth, how can the focus and the context be rendered at interactive rate.

To solve the four questions computationally, we restate them in light of computational resources:

1) For exploring internal structure in view that a neuro-surgeon has in an operating room we should somehow remove the occluding voxels in layers, as detailed in [1]. Once the proposed algorithm presents artifacts that may prevent a thorough visual analysis, we ask ourselves how to attenuate or to completely remove these artifacts.
2) As we consider the focal region the intersection of 3D volume dataset and a query geometry, i.e. volumetric probe and movable magnifying lens (Section I-B), we may equivalently ask how to map a cursor location on the screen (2D) onto the centroid of the query geometry in the data domain (3D).
3) In order to preserve the original information in the focal region, we decide to assign to a pixel only the color of the closest visible voxel. Hence, we may put the question in the following way: how to classify all the voxels according to their depth, from which we may decide the region to be effectively clipped away.
4) Once the visual distinction between the focus and the context relies on the rendering technique, on the transfer function, and on the projection parameters, the question is, in fact, how to efficiently combine them for a diagnosis oriented visualization.

C. Solutions

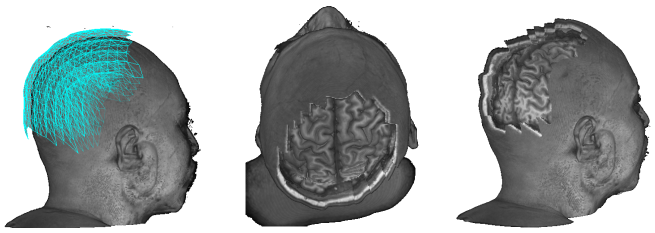
In this section we proceed to our algorithmic solution for each issue.

Curvilinear Reformatting: In comparison with [1], we refine the mesh offset algorithm, using a new way to lead the degenerates faces, and improve the voxelization algorithm of the crop solver presented in Fig. 4.

For offsetting, the mesh vertices are displaced in their normal direction as explained in section I-A, until they become degenerate. We consider a face degenerate when its triangle-area-to-largest-side-length ratio is smaller than a pre-specified value. The decimation algorithm proposed in [21] is applied to remove the degenerate faces. A greedy strategy is used to decide which face to be collapsed. New mesh vertices are further displaced in their normal direction toward the brain's midpoint. The procedure is repeated till the pre-defined depth is reached.

To remove the voxels for revealing the tissue at a specified depth, we should perform voxelization. Voxelization is often performance critical. Many GPU-based algorithms have been proposed for improving its performance [22]. To our knowledge, all of them are view-dependent, i.e. the sampling rate is based on the screen resolution where the object of interest is displayed. We modify slightly the GPU-based algorithm proposed in [23] and render images with resolution similar to that of the data volume in offscreen mode.

Fig. 7 illustrates the result of our algorithm applied on the same head presented in Fig. 2. Fig. 7(a) shows a series of offset meshes parallel to the scalp. From Fig. 7(b) and Fig. 7(c), two distinct views of the cropped head, we may observe that the artifacts have been removed.



(a) Parallel layers (b) Top view (c) Side view

Fig. 7. Cropping.

2D to 3D mapping: As mentioned in Section II, we apply the algorithm presented in [17] to restric the mouse movement on the visible voxels that are closest to the user. The required volume depth map is read from the offscreen frame buffer back to CPU. This suffices for brushing ROI (Fig. 4) and for positioning a *movable lens* on the visible voxels (Fig. 6).

If free spatial cursor movement is needed, as for positioning the center C of a *volumetric probe* (Fig. 5), new strategy should be devised. We decompose a spatial movement (x, y, z) into two planar movements (x, y) and (x, z). To distinguish these two motion modes, we use Z key. When this key is not pressed, we consider that z is fixed in z_0 and map the coordinates (x, y) of the pixel, over which the cursor pointer hovers, onto 3D coordinates (x, y, z_0). When the Z key is held pressed, we estimate Δz from the variations of cursor's device coordinates, Δx and Δy, and the diagonal display size H, i.e. $\Delta z = \frac{\sqrt{\Delta^2 x + \Delta^2 y}}{4H}$. If the cursor's movement is upwards, we map (x, y) onto $(x, y, z + \Delta z)$; otherwise, it is mapped onto $(x, y, z - \Delta z)$.

Focal region: To select the voxels inside the convex volumetric probe with radius r, we devise a GPU-based implementation for the procedure presented by Zhou *et al.* [3]. The probe geometry is offscreen rendered on GPU, as shown in Fig. 5, to generate two depth maps: one of the front

face and another of the back face. With these two maps, we select all the voxels z_{voxel} along each viewing ray that satisfies simultaneously the conditions $(z_{voxel} \geq z_{FrontFace})$ and $(z_{voxel} \leq z_{BackFace})$, where $z_{FrontFace}$ and $z_{BackFace}$ are the values in the depth map. In Fig. 8 the front face's depth map is colored in red and the back face in magenta. All voxels between them, colored in green, belong to the focal region and the rest of voxels inside the proxy geometry, in orange, belong to the contextual region.

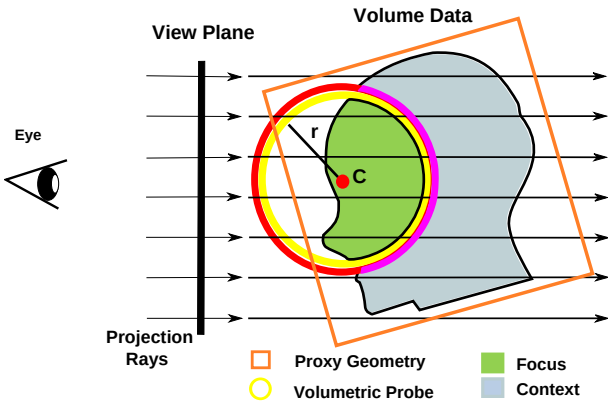


Fig. 8. Volumetric probe.

Our movable lens consists of a disk, with radius LR and center LC positioned on the visible volume data (x, y, z). LC is represented by a red point in Fig. 9. The lens geometry is also offscreen rendered, as presented in Fig. 6, to generate its depth map for controlling the ray traversal direction. If the viewing ray is inside the lens disk, it converges to a focal point FP; otherwise it keeps parallel. Fig. 9 illustrates two projection modes. Our magnification procedure is similar to the one proposed by Wang *et al.* [5].

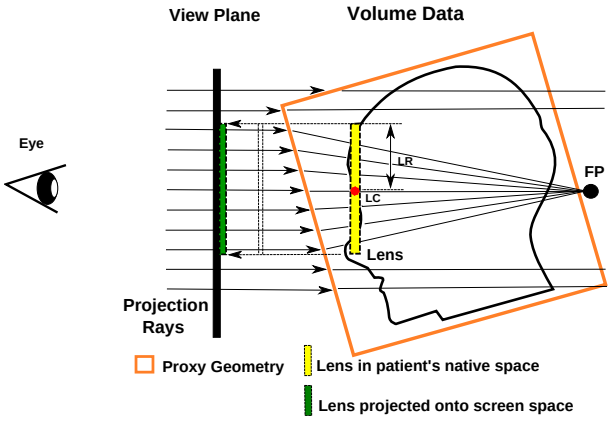


Fig. 9. Magnifying lens.

Integration: To improve the contrast enhancement for an specific region of interest, we propose to keep two 1D transfer functions, one for the focal region delimited by a query geometry and the other for the remaining voxels. In this way, we may use all the range of intensity values to distinguish the scalar values in the region of interest without loss of reference. Fig. 10 presents the visualization of the same volume dataset with two distinct transfer functions. Observe that through an appropriate setting the amygdala, referenced by (1), and hippocampus, indicated by (2), are distinguishable in Fig. 10(b), while in Fig. 10(a) two structures are almost imperceptible. In both cases, the context is preserved.

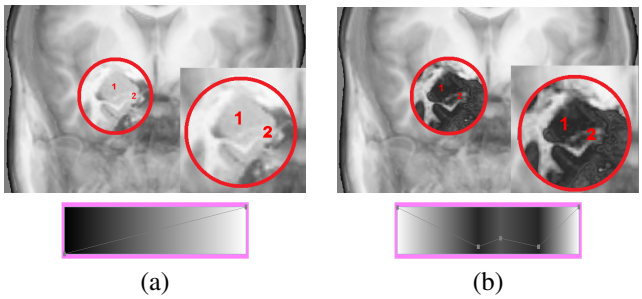


Fig. 10. Contrast enhancement: (a) context monotonic transfer functions, and (b) focal non-monotonic transfer functions.

IV. Implementation

Since our proposed dynamic query tools are designed on the basis of ray shooting, except the curvilinear reformatting, it is straightforward to integrate them into a GPU-based ray-casting architecture [18]. In this section we present their implementation details.

A. Crop solver

With respect to the algorithm proposed in [1], our crop solver is improved by applying the algorithm proposed in [21] to remove degenerate faces. This algorithm is implemented in the open-source OpenMesh [24]. We simply replaced the older degenerate face removal codes by the adapted functions available in OpenMesh.

B. The volumetric probe

Whenever the center of the probe geometry is changed, it is rendered twice into a frame buffer object in order to get the two depth maps from the probe faces along the projection direction, as shown in Fig. 5. These two depth maps are bound as 2D textures in GPU memory and used by the onscreen ray-casting volume rendering shader to correctly select the transfer functions.

Only a slight modification is necessary in a single-pass ray-casting rendering fragment shader to select the voxels inside the probe. From each pixel (x_{sc}, y_{sc}) a single ray is cast into the 3D volume data and the ray is sampled at discrete positions (x, y, z). While (x, y, z) is in the interior of the volume data, the depth of each voxel along the ray is computed. This depth is tested against the extremes of the interval that is in the interior of the probe geometry. If it is the case, the focal transfer function is applied to assign the color and the opacity to the current voxel; otherwise, the context transfer function is used.

C. The movable magnifying lens

A distinguishing feature of our proposed lens is its capability to follow the mouse motion smoothly. Whenever the 2D cursor pointer is changed, the lens center LC is displaced to it. It is achieved by copying back to CPU the offscreen rendered volume depth map (Fig. 5) which is used to 2D–3D mapping. Nevertheless, as already explained in Section III-C, the perception that the lens slides smoothly over the visible data comes indeed from the fact that we use the lens geometry to select projection mode in a ray-casting volume rendering shader. For this reason, analogously to the probe geometry processing, the lens geometry is offscreen rendered and bound as a 2D texture in video memory.

Again, only a slight modification is necessary in a single-pass ray-casting rendering fragment shader. In the focal regions, the perspective viewing direction is computed. We traverse along this direction until we find a visible voxel. The intensity value of this voxel is mapped to optical properties through the focal transfer function.

V. Results and Discussion

The evaluation platform was a desktop Intel®Core2 Duo E7500 2.936 GHz CPU with 2GB RAM and a NVIDIA GeForce GT240, 1GB VRAM [25]. Patient data were acquired by a RM 3T Philips Intera-Achieva Scanner at our university hospital and have a volume size of 240×240×180, 12 bits. Aiming to ensure coherence size between data volume and query tools, we normalize the radius of the probe and the lens to [0,1], then their diameters keeps correspondence with the longest side of the data volume. The clock() function available in C++ was used to measure the performance time: it is called before and after the procedure whose processing time we want to measure and we consider the quotient of the difference of the returned values in ticks and the constant CLOCKS_PER_SEC as the processing time in seconds.

A. Performance

In Fig. 11, we present the time performance of a classical single-pass GPU-based raycasting volume renderer and its modified version that integrates our dynamic query tools. We compute the objects depth maps in offscreen mode and render the volume data in onscreen mode. For small output resolutions the performance drop is negligible. Even when the output size is increased, drop does not exceed 20%.

Observe that apparently the probe mode demands more computational resources. In the full screen mode ($\approx 1024^2$), it seems to present the worst performance. Nevertheless, for appropriate visual feedback during displace2DLens interactions as shown in Fig. 6, only one tool geometry rendering and one complete scene rendering is necessary. This takes approximately $\frac{1}{61} + \frac{1}{46}$ = 0.038s, or 26 FPS, which is an acceptable interactive rate. On the other hand, in the probe interaction mode, schematised as displace2DProbe in Fig. 5, we need two offscreen renderings of tool geometry and one complete onscreen scene rendering. This amounts to $2.\frac{1}{61} + \frac{1}{48}$ = 0.054s, or 19 FPS. Even though, it is still further under the response time limits for having the user feel that the system is reacting immediately [26].

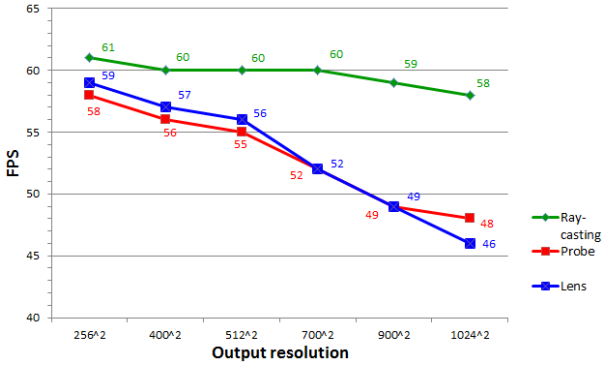


Fig. 11. Performance results in FPS

B. Usability

To test the usability of the tool, a preliminary evaluation was done with clinical MRI experts. We had six representative volunteers: four neuroscientists from the Laboratory of Neuroimaging at our university and two radiologists from another hospital. In this preliminary analysis, we chose to study MR images of patients with epilepsy associated with focal cortical dysplasias (FCD). FCD is a type of malformation of cortical development often associated with seizures refractory to antiepileptic drug [27]. Patients therefore are often investigated for the possibility of surgical treatment with the aim of the resection of the dysplastic lesion. It is know that the optimal result of the surgical treatment is only achieved with the complete resection of the lesion [28]. In this context, the accurate identification of the FCD in the pre-operatory MRI is vital for the surgical success. According to the recent histological classification and neuroimaging studies, the majority of FCDs type IIb (presence of dysmorphic neurons and baloon cells) can be visually identified in MR images by epilepsy experts. Differently, FCDs type IIa (presence of dysmorphic neurons without baloon cells) are often missed in the conventional visual analysis [29]. Subtle MRI signs as cortical thickening, abnormal gyri, and poor delineation of the transition between white and gray matter are the main findings of FCD type IIa [30] and the detection of this type of lesion is often a challenge.

In our preliminary experiment, images of 4 patients with FCD type IIa and 2 FCD type IIb localized by the combination of clinical, eletroencephalography and visual MRI evaluation were selected. The MRI experts were then asked to blinded localize the lesion. No information was given to the volunteers except that the patient had FCD. The two radiologists and one of the neuroscientists refused the task. The argument of these specialists was that due to the subtle abnormalities observed in MRIs of patients with FCDs, this type of lesion is always evaluated together with the clinical and electroencephalographic information. The other three experts agreed with the task.

Among the three volunteers that realized the tasks, there was 83% concordance in localizing the subtle brain lesions.

C. Limitations

A shortcoming is that our proposed query tools are highly dependent on the hardware-accelerated graphics resources to achieve interactive frame rates. GPUs with at least 1GB VRAM are required.

The decimation algorithm has, in the worst case, time complexity $O(v^2)$, where v is the number of vertices. This implies that it cannot satisfy the interactivity requirement if the number of mesh vertices is huge. Fortunately, we are working with the meshes that have less than 3000 vertices.

VI. Conclusion

In this paper, we introduced two dynamic query tools that may aid the physicians to seek for brain structural abnormalities in a focus+context visualization environment. We present a solution for making it more interactive and easy-to-manipulate with a conventional 2D mouse. Together with a flexible cropping tool, that we improved, we conjecture that our tools can assist neuroscientists to discover subtle lesions that are visually distinguishable.

The key to our proposed paradigm is to improve the responsiveness of each displayed voxel, so that it may promptly "react" under the user actions. We showed how we may explore the current graphics resources to enhance the voxel's responsiveness. Depth maps and pre-processed depth control volume play an important role. In nutshell, we make extensive use of FBO to generate depth images which provide us the missing z-coordinates. Then, it becomes feasible to treat the 2D pixels as 3D spatial samples at interactive rate.

Although the usability test is preliminary, the agreement rate was high what points toward a good usability of the tool. New validation tasks will be conducted shortly. To surpass the issue of the high percentage of volunteers that refused to perform the tasks, the new validation tasks will be carried out in the light of the clinical and electroencephalographic context and a comparison of the agreement rates with other neuroimaging techniques will be performed. As a mid-term goal we would like to integrate our proposed tools in an interactive multimodal visualization environment.

Acknowledgment

References

[1] S.-T. Wu, C. L. Yasuda, and F. Cendes, "Interactive curvilinear reformatting in native space," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 2, pp. 299–308, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.40

[2] H. Hauser, L. Mroz, G. Italo Bischi, and M. Groller, "Two-level volume rendering," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 7, no. 3, pp. 242–252, july-sept. 2001.

[3] J. Zhou, M. Hinz, and K. Tönnies, "Focal region-guided feature-based volume rendering," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, 2002, pp. 87–90.

[4] E. LaMar, B. Hamann, and K. Joy, "A magnification lens for interactive volume visualization," in *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, 2001, pp. 223–232.

[5] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman, "The magic volume lens: an interactive focus+context technique for volume rendering," in *Visualization, 2005. VIS 05. IEEE*, oct. 2005, pp. 367–374.

[6] R. Mullick, R. N. Bryan, and J. Butman, "Confocal volume rendering: fast, segmentation-free visualization of internal structures," in *Proc. SPIE 3976*. Medical Imaging 2000: Image Display and Visualization, April 2000.

[7] C. Roden and M. Brett, "Stereotaxic display of brain lesions," *Behavioural Neurology*, vol. 12, pp. 191–200, 2000.

[8] D. Weiskopf, K. Engel, and T. Ertl, "Interactive clipping techniques for texture-based volume visualization and volume shading," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 3, pp. 298–312, july-sept. 2003.

[9] R. Huff, C. A. Dietrich, L. P. Nedel, C. M. D. S. Freitas, J. L. D. Comba, and S. D. Olabarriaga, "Erasing, digging and clipping in volumetric datasets with one or two hands," in *VRCIA*, 2006, pp. 271–278.

[10] S. Bruckner and M. E. Gröller, "Exploded views for volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1077–1084, 9 2006. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2006/bruckner-2006-EVV/

[11] I. Viola, A. Kanitsar, and M. Gröller, "Importance-driven feature enhancement in volume visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 11, no. 4, pp. 408–418, july-aug. 2005.

[12] S. Bruckner and M. E. Gröller, "Volumeshop: An interactive system for direct volume illustration," in *Proceedings of IEEE Visualization 2005*, H. R. C. T. Silva, E. Gröller, Ed., oct 2005, pp. 671–678. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2005/bruckner-2005-VIS/

[13] P. Sikachev, P. Rautek, S. Bruckner, and M. E. Gröller, "Dynamic focus+context for volume rendering," in *Proceedings of Vision, Modeling and Visualization 2010*, University of Siegen, Germany, 2010, pp. 331–338.

[14] M. Cohen, K. W. Brodlie, and N. Phillips, "The volume in focus: hardware-assisted focus and context effects for volume visualization," in *SAC*, 2008, pp. 1231–1235.

[15] S.-J. Kim and M.-Y. Yang, "Triangular mesh offset for generalized cutter," *Comput. Aided Des.*, vol. 37, no. 10, pp. 999–1014, Sep. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.cad.2004.10.002

[16] H. C. Batagelo and S.-T. Wu, "What you see is what you snap: snapping to geometry deformed on the GPU," in *Proceedings of the 2005 Symposium on Interactive 3D Graphics, SI3D 2005*, Washington, DC, USA, 2005, pp. 81–86.

[17] S.-T. Wu, J. E. Yauri Vidalon, and L. de Souza Watanabe, "Snapping a cursor on volume data," in *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, aug. 2011, pp. 109–116.

[18] K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, and D. Weiskopf, *Real-Time Volume Graphics*. AK Peters, 2006.

[19] OpenGL, "OpenGL, the cross-platform graphics API," 2010, [Acessado em Agosto 2011]. [Online]. Available: http://www.opengl.org

[20] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. London: Academic Press, 1999.

[21] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 209–216. [Online]. Available: http://dx.doi.org/10.1145/258734.258849

[22] M. Schwarz and H.-P. Seidel, "Fast parallel surface and solid voxelization on gpus," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 179:1–179:10, dec 2010. [Online]. Available: http://doi.acm.org/10.1145/1882261.1866201

[23] E.-A. Karabassi, G. Papaioannou, and T. Theoharis, "A fast depth-buffer-based voxelization algorithm," *J. Graph. Tools*, vol. 4, no. 4, pp. 5–10, dec 1999. [Online]. Available: http://dl.acm.org/citation.cfm?id=643328.643330

[24] R. A. Computer Graphics Group. (2013) Openmesh. Accessed April 2013]. [Online]. Available: http://http://www.openmesh.org/

[25] J. E. Yauri Vidalón, "VMTK: Visual Manipulation Tool for 3D MR Neuro-Images," 2012, [acessed in June 2013]. [Online]. Available: http://www.dca.fee.unicamp.br/projects/mtk/vidalon.html

[26] N. N. Group, "Response times: The 3 important limits," accessed in April 2013. [Online]. Available: http://www.nngroup.com/articles/response-times-3-important-limits/

[27] I. Blümcke, M. Thom, E. Aronica, D. D. Armstrong, H. V. Vinters, A. Palmini, T. S. Jacques, G. Avanzini, Barkovich, G. Battaglia, A. J. Becker, C. Cepeda, F. Cendes, N. Colombo, P. Crino, J. H. Cross, O. Delalande, F. Dubeau, J. Duncan, R. Guerrini, P. Kahane, G. Mathern, I. Najm, C. Ozkara, C. Raybaud, A. Represa, S. N. Roper, N. Salamon, A. Schulze-Bonhage, L. Tassi, A. Vezzani, and R. Spreafico, "The clinicopathologic spectrum of focal cortical dysplasias: a consensus classification proposed by an ad hoc Task Force of the ILAE Diagnostic Methods Commission." *Epilepsia*, vol. 52, no. 1, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3058866&tool=pmcentrez&rendertype=abstract

[28] P. Krsek, B. Maton, P. Jayakar, P. Dean, B. Korman, G. Rey, C. Dunoyer, E. Pacheco-Jacome, G. Morrison, J. Ragheb, H. V. Vinters, T. Resnick, and M. Duchowny, "Incomplete resection of focal cortical dysplasia is the main predictor of poor postsurgical outcome." *Neurology*, vol. 72, no. 3, pp. 217–23, 2009. [Online]. Available: http://www.biomedsearch.com/nih/Incomplete-resection-focal-cortical-dysplasia/19005171.html

[29] N. Colombo, L. Tassi, F. Deleo, A. Citterio, M. Bramerio, R. Mai, I. Sartori, F. Cardinale, G. Lo Russo, and R. Spreafico, "Focal cortical dysplasia type iia and iib: Mri aspects in 118 cases proven by histopathology." *Neuroradiology*, 2012.

[30] F. Cendes, "Neuroimaging in investigation of patients with epilepsy," *CONTINUUM: Lifelong Learning in Neurology*, vol. 19, no. 3, Epilepsy, pp. 623–642, 2013.