Mak WH, Wu Y, Chan MY *et al.* Visibility-aware direct volume rendering. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(2): 217–228 Mar. 2011. DOI 10.1007/s11390-011-9428-3

Visibility-Aware Direct Volume Rendering

Wai-Ho Mak[1] (麦伟豪), Yingcai Wu (巫英才)[2,*], Ming-Yuen Chan[1] (陈明远)
and Huamin Qu[1] (屈华民), *Member, IEEE*

[1] *Department of Computer Science and Engineering, The Hong Kong University of Science and Technology Hong Kong, China*

[2] *Department of Computer Science, The University of California, Davis, CA, U.S.A.*

E-mail: {wallacem, pazuchan, huamin}@cse.ust.hk; ycwu@ucdavis.edu

Received July 24, 2009; revised November 28, 2010.

Abstract Direct volume rendering (DVR) is a powerful visualization technique which allows users to effectively explore and study volumetric datasets. Different transparency settings can be flexibly assigned to different structures such that some valuable information can be revealed in direct volume rendered images (DVRIs). However, end-users often feel that some risks are always associated with DVR because they do not know whether any important information is missing from the transparent regions of DVRIs. In this paper, we investigate how to semi-automatically generate a set of DVRIs and also an animation which can reveal information missed in the original DVRIs and meanwhile satisfy some image quality criteria such as coherence. A complete framework is developed to tackle various problems related to the generation and quality evaluation of visibility-aware DVRIs and animations. Our technique can reduce the risk of using direct volume rendering and thus boost the confidence of users in volume rendering systems.

Keywords visualization systems and software, volume visualization

1 Introduction

Volume visualization enables people to explore and study volumetric data by using interactive computer graphics techniques. Among the many volume visualization techniques, direct volume rendering (DVR) is best known for its flexibility. With the controllable transparency and color settings of voxels, DVR provides an informative 2D view of volumetric data revealing the relationship among different 3D structures, and also helping users gain insights into the data. The mapping of density values to transparency and color is determined by transfer functions which are core parts of DVR.

Over the last two decades, many researchers have worked on various related problems of direct volume rendering. In particular, much work has been conducted on transfer function design. Also, tremendous efforts have been put into improving efficiency and interface. Most research work focuses on detection of features in the data. In most commercial systems, some preset transfer functions are used to ease the difficulties of using DVR for non-expert users. The time performance of DVR has also been dramatically improved and there already exist some intuitive interfaces for interactive volume exploration.

While many technical issues of direct volume rendering have been studied extensively, this sophisticated technique is not yet being widely used in practice. Usability is one main issue which has already attracted much attention from the visualization community. Another major obstacle, which currently has not drawn adequate attention, is that users are unsure of the safety aspect of using this flexible technique in volume exploration. Because there are often multiple structures and transparency is used in direct volume rendered images, some structures or parts of the structures may become over transparent or hidden, users may not be aware of the existence of those structures or parts in the rendered images. During the whole visualization process, there is no guarantee that all or most of the information has been effectively revealed. Some important information may be missing as some voxels can be assigned with very low opacity during the entire volume exploration process. Even if some voxels are given high opacity values, they may not be well revealed due to the occlusion. As a result, a certain degree of risk of missing important information is always associated with DVR.

Regular Paper
The work is supported in part by Hong Kong RGC CERG under Grant No. 618705.
*Corresponding Author

The safety issue is critical for a wider adoption of direct volume rendering in practice. However, to the best of our knowledge, it has not been well addressed before.

There are some trivial approaches for this problem. For example, we can directly give all the invisible voxels high opacity values, but obviously it cannot really solve the problem because some voxels may still be invisible due to occlusion. Another possibility is that we gradually remove voxels with high visibility, then the hidden voxels may be eventually revealed. But since too much context is removed, users may only see a set of voxels from which no structure-level information can be identified. All these trivial approaches would only generate messy direct volume rendered images that are not informative and cannot reveal missing information well.

In this paper we investigate a systematic way to present missing information in a region of interest (ROI) to users. We assume that some preset, user-tuned, or automatically generated transfer functions have been used to reveal some information in the data. Viewpoint, lighting, and structure colors are assumed to be already carefully chosen. Users can use our system to generate a set of DVRIs, which can then be used to compose a comprehensive animation, that effectively reveal the missing information in the ROI.

The generated DVRIs should meet the following requirements:

• *Guaranteed Visibility*. Interesting structures are guaranteed to have certain level of visibility in the original DVRIs given by users or the DVRIs generated by our method.

• *Coherent Information*. The presentation of the structures is coherent with the actual structures so that the shapes of the interesting structures can be clearly and correctly perceived.

• *Familiar Context*. The DVRIs and the original DVRIs share some common content so that users are oriented about the spatial relation among the structures presented in them.

The major contributions of this paper are:

• We investigate one weakness of DVR and propose a semi-automatic solution as a means to increase the safety of volume visualization with direct volume rendering. We introduce a complete framework that is capable of detecting and revealing the missing information.

• We present visibility metric, coherence metric, and context metric to access the quality of a semi-automatically generated DVRI using our framework.

• We systematically attack the optimization problem involved in generating DVRIs according to the aforementioned quality metrics.

• We demonstrate how our framework can be extended as a volume exploration method. Our method can gradually generate new DVRIs based on what have already been revealed.

This paper is organized as follows. After introducing related work in Section 2, we describe our framework in Section 3. The preprocessing procedures of our system are then described in Section 4. Section 5 introduces the quality metrics, followed by Section 6, in which we explain the rest of the semi-automatic DVRI generation. Production of animation based on the generated DVRIs is introduced in Section 7. In Section 8, we present experimental results to demonstrate our proposed framework. Finally, discussion is given in Section 9 followed by conclusion in Section 10.

2 Related Work

Direct volume rendering has been thoroughly investigated in the visualization field and many papers have been published on transfer function design and efficient rendering. Some recent results have been summarized in several books and a few survey papers. In this section, we only review some closely related work.

2.1 Transfer Function Design

Searching for a proper transfer function is a challenging problem. He *et al.*[1] used a stochastic search technique to explore the transfer function parameter space. A semi-automatic transfer function generation method based on the histogram volume is proposed in [2]. Kniss *et al.*[3] proposed several useful manipulation widgets for transfer function design. A set of tools for interactive specification of parameters for volume visualization system built on VolumePro technology was introduced by König *et al.*[4] Tzeng *et al.*[5] proposed a classification method based on user painting. Very recently, visibility becomes a major component in transfer function design[6], as proposed by Correa and Ma. In our framework, we use a divide-and-conquer approach for transfer function design to achieve our various purposes. We also introduce local depth opacity-modulation as a voxel-level visibility boosting technique.

2.2 Genetic Algorithm

Genetic algorithm (GA) is a widely-used technique for searching solutions to optimization problems[7]. He *et al.*[1] first employed GA for transfer function design. GA was also exploited to optimize visualization quality through automatic visualization parameters selection[8]. Wu and Qu[9] developed a framework for editing DVRIs using GA with the energy function based on image similarity. In this paper, we also use GA for generating DVRIs but we have different purposes.

Also, our energy function is formulated particularly for the image quality and visibility guarantee.

2.3 Quality Metrics

Quality measurement is a fundamental problem in computer vision. Image quality can be measured by objective metrics and subjective metrics. The subjective metrics, such as mean option score (MOS) which has long been considered as the best image quality measurement, are too expensive to be useful in practice as they require a number of human observers. The objective image quality measures can be generally classified into full-reference[10], no-reference[11], and reduced-reference[12] measures. For details, interested readers can refer to [13]. Wang *et al.*[14] proposed an image-based quality metric which measures the contribution of the multiresolution data blocks to the resulting image for their multiresolution level-of-detail selection and rendering framework. They further developed a reduced-reference method for volumetric data quality measurement[15]. However, the study of quality metric for DVRIs is scarce. Our framework needs to estimate the image quality, as we require that our resulting DVRIs should not only reveal the missing information but also be coherent and have enough contextual information. Consequently, our image quality metrics are built from the perspective of the coherence and context criteria, which are different from previous work.

2.4 Coherence

Coherence often indicates the logical and orderly consistent relationship of items. Gröller and Purgathofer provided an excellent survey on coherence in computer graphics[16]. Among various types of coherence discussed in their survey, object coherence and image coherence are most relevant to our work. Object coherence states that local close regions in 3D spatial domain are likely to be occupied by similar entities, which is also employed by Lundstrom *et al.*[17] for feature detection. Image coherence is similarly defined in the 2D image domain. In addition, 3D objects and their corresponding 2D projections should have similar degrees of connectedness and smoothness.

2.5 Feature Revealing Techniques

GPU-accelerated interactive clipping methods proposed by Weiskopf *et al.*[18] are useful for data exploration. A novel importance-based approach was developed by Viola *et al.*[19] for showing the underlying features in a "focus + context" manner. Krüger *et al.*[20] and Bruckner *et al.*[21] introduced new context preserving visualization techniques for interactive visualization of volumetric data. Rezk-Salama and Kolb[22] proposed a technique called opacity peeling to reveal hidden structures by removing visible voxels layer by layer. Our approach also reveals underlying features, yet in addition, it ensures that all voxels of features become visible in a set of DVRIs.

3 System Overview

We assume that the volumetric dataset is already segmented and a set of DVRIs are already generated by users. In the preprocessing stage, the segmented structures are decomposed into components called atomic structures. Their properties and the relation among them are stored. After the volume decomposition, the system analyzes and records the visibility and visualization quality of each atomic structure in all the given DVRIs. This information is tracked and updated throughout the whole DVRI generation process. After preprocessing, users will define the ROI by various means provided by our system, e.g., choosing intensity ranges, directly selecting structures from the segmentation result set given structure thumbnail images, using a graph interface, or brushing on the DVRI to indicate certain interested areas. Our system is also capable of detecting the region of interest automatically by analyzing the source DVRIs, the visualization history, and some optional user input. A set of DVRIs will then be automatically generated to reveal missing information in the ROI effectively. We use genetic algorithm to search for a good opacity seed value for each atomic structure in every generated DVRI. Given an opacity seed value, a 2D transfer function is then automatically generated for the corresponding atomic structure. With different 2D transfer functions being assigned to all the atomic structures in the volume, an intermediate DVRI is rendered and evaluated with the energy function used in the genetic algorithm. Our energy function consists of three metrics, namely the visibility metric, the coherence metric, and the context metric. These three metrics respectively guarantee visibility of unrevealed voxels, maintain the coherence of the presented information, and add in a certain amount of shared context so that inter-structure relation can be easily understood. When the evaluated quality of an intermediate DVRI is good enough according to users' needs, i.e., user-defined thresholds, it is output as one of the generated DVRIs. Every such successful generation is subsequently followed by an update to the structure visibility and quality information. The DVRI generation process is repeated until all structures of interest become visible and each atomic structure in the ROI is revealed in high quality at least once. Finally, a subset of the generated DVRIs can be chosen to generate an animation that gives a comprehensive presentation of the structures of interest.

4 Data and DVRI Preprocessing

4.1 Decomposition

Given the segmentation, the voxels in the volume are grouped into different structures. The resulting segments, however, are sometimes too large to receive enough visibility when they are rendered due to serious self-occlusion at the voxel level. Therefore, we need to further break those segments down into smaller pieces. The decomposition process subdivides a segment until it can receive enough visibilities when it is rendered individually without the presence of other segments. The subdivision always iteratively produces two clusters of equal sizes in order to prevent a segment becoming too many small fragments. The clustering can be based on one or more attributes. Our current approach is to use voxel depth values, i.e., distance to the view-plane, to do the partitioning. Voxels are sorted according to their depth values and then divided into two clusters of the same sizes. The resulting segments are atomic structures which are small clusters of voxels which belong to the same structure.

4.2 Transfer Function Design for Atomic Structure

For each evolution step in the genetic algorithm, new candidate solutions are generated. This involves generating an intermediate DVRI given an atomic structure opacity seed array returned by the genetic algorithm. There are three important considerations for designing a good transfer function for an atomic structure. First, a certain percentage of the voxels in the atomic structure must have a certain level of visibility when the structure is not occluded by other structures even if there is self-occlusion among its voxels. Second, the information in an atomic structure should be conveyed effectively. Third, the transfer function has to be generated automatically and efficiently as it has to be generated for each atomic structure before each evaluation. To address all three concerns, we develop an efficient opacity transfer function for all the voxels in an atomic structure. While we adopt the widely used gradient magnitude opacity-modulation, we introduce a new technique called depth opacity-modulation. The opacity value for voxel v in an atomic structure s is:

$$\alpha_v = \begin{cases} Z(v), & \text{if } Z(v) \leqslant 1, \\ 1, & \text{otherwise,} \end{cases}$$
$$Z(v) = \alpha_s \cdot G(g_v) \cdot r^{d_v - d_{\min}(s)}$$

where α_s is the given seed opacity value for the atomic structure, g_v is the gradient magnitude of v, and G is a monotonically-increasing gradient magnitude opacity-modulation function which maps gradient magnitudes to opacity-modulation factors. d_v is the depth value of v, $d_{\min}(s)$ is the minimum depth of all voxels in the atomic structure s, and r is the depth opacity-modulation rate which controls the influence of voxel's depth in modulation. As α_s, G, r, and $d_{\min_s}$ are all the same for voxels in the same atomic structure, this formulation is essentially a 2D transfer function mapping from voxel depth and gradient magnitude to opacity. This function gives high opacity values to voxels of high depth value (i.e., at the back side of the atomic structure with respect to the viewpoint). It ensures that all of those voxels will still receive enough visibility even if they are occluded by voxels of low depth value (i.e., at the front side of the atomic structure). Depth modulation can significantly reduce the number of atomic structures as structure visibility guarantee can now be achieved much more easily. Moreover, the gradient modulation transfer function G applies large opacity-modulation factors to voxels with high gradient values and small opacity-modulation factors to voxels with low gradient values. It therefore enhances the clarity of the details in an atomic structure and improves effectiveness of information delivery as regions with more intensity variation are often of more interest to users.

4.3 Structure-Based Analysis of DVRIs

After decomposition of the data, a set of atomic structures are generated as the basis for analysis. For each DVRI, we first find out which structure is presented in the DVRI. Then for each of the presented structures, we further analyze their quality with the metrics described in Section 5. The unclear or missing structures in the region of interest will be the foci of the DVRIs to be generated.

4.4 Region of Interest Selection

Not all structures are of interest to the users as different users have different visualization goals which result in different regions of interest in the volume. The simplest ROI selection is achieved by selecting one or more interested data value (intensity) ranges. This will work for simple volume datasets in which classes of structures can be easily specified by just data values. In addition, as segmentation is assumed to be performed, users can also directly select the structures of interest from the segment gallery. Moreover, users can specify ROI on DVRI by brushing on it. After brushing, our system will list all the segmented structures that are hit by rays casted from the brushed pixels for user selection of interesting structures.

5 Quality Evaluation Metrics

To obtain high-quality DVRIs that can effectively convey the missing information, we need some metrics to measure their qualities. In this section, we introduce three metrics used in our system for DVRI quality evaluation.

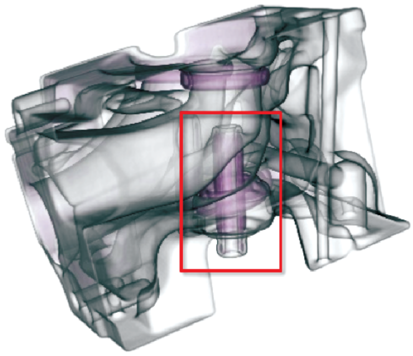
5.1 Visibility Metric

A voxel is said to be visible when its visibility is larger than a certain voxel-level visibility threshold. As voxels of high gradient magnitude are often of higher importance than voxels of low gradient magnitude as they represent feature outlines, the voxel-level visibility threshold should depend on the voxel's gradient magnitude. While the threshold base value t_{vv} is specified by the users, it is first modulated by the gradient magnitude opacity-modulation function G, which is mentioned in Subsection 4.2, when it is applied to check if voxel v is visible. Therefore, the voxel-level visibility threshold used by our system for v is $t_{vv} \cdot G(g_v)$. Based on that, the structure visibility of an atomic structure s is defined as $V(s) = \frac{N_{vv}(s)}{N_{tv}(s)}$ where $N_{vv}(s)$ is the number of visible voxels in s while $N_{tv}(s)$ is the total number of voxels. It measures the percentage of visible voxels in s. If it is equal to or larger than the structure visibility guarantee threshold t_{sv}, then the corresponding atomic structure is considered as visible enough.

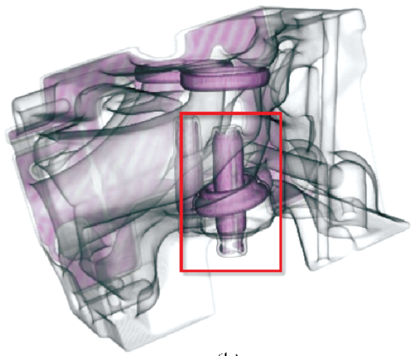
5.2 Coherence Metric

Using only the visibility metric, we can already generate a set of DVRIs to have the missing information revealed and to provide interesting structure visibility guarantee. The missing information, however, is unlikely to be well-revealed if it is just randomly arranged with the surrounding context. Therefore we propose to use a coherence metric to help arrange the presentation of information so that the missing information is presented in a comprehensible and clear way.

In a 2D image, for users to perceive the shape of a structure, the structure shading variation on the image is a very important clue. However, in a DVRI, as transparency is used, partial occlusion can often occur and such occlusion can distort the structure shading variation. Hence, for a structure to be correctly and clearly revealed, its shading distribution on the image should not be distorted too much by occlusion. Fig.1 illustrates this coherence issue that should be considered. In Fig.1(a), the coherence of an engine part is destroyed as there is too much occlusion on the structure causing unclear appearance even though it is visible in the image to some extent. It also looks like as if it is of several fragments because of its inconsistent appearance in different regions due to different degrees of occlusion. After adjusting the opacity setting, the coherence of that engine part is restored and thus clearly shown in Fig.1(b). Our coherence metric checks whether a structure is presented correctly and clearly on a DVRI.



(a)



(b)

Fig.1. Coherence problem. (a) A DVRI showing an engine part (enclosed by a red rectangle) shown with bad coherence. While it is visible, it is unclear in the image due to too much non-uniform occlusion. (b) A DVRI showing the same part, which is displayed clearly, with good coherence.

The coherence evaluation is conducted on a focus structure, which is unique in a candidate solution, in a generated DVRI. Before actually computing the coherence of the structure, we need to check if it is visible enough in the image according to the previously mentioned visibility metric. Being visible in the image is a necessary condition, therefore, if the structure is not visible enough in the image, the coherence is considered as bad automatically. In that case, the algorithm returns $(1 + (t_{sv} - V(f))/t_{sv}) \cdot K_h$, where K_h is a large constant, to represent bad coherence quality. The worse the visibility, the worse the coherence. Assume the structure is visible enough in the image, i.e., $V(f) \geqslant t_{sv}$, to compute the coherence metric, we first exclusively render the focus structure and save the grayscale image of the rendered image to capture the luminance variation (i.e., shading distribution) of the focus structure projection on the image. Then, we analyze the difference image $image_{diff}$ of its grayscale image and that of the given DVRI. The statistical variance σ^2 of the difference values in its pixels is used as an estimation of structure coherence. Similar difference values in the pixels give small variance and thus imply high structure coherence as the shading distribution is generally preserved. On the other hand, large variance implies inconsistent change to the structure appearance, which means low structure coherence. Note that uniform change of the shading distribution of a structure projection will give zero variance of the difference values. In that case, the coherence is good because the shading distribution is not really disturbed. The algorithm for calculating the coherence metric $H(c)$ given a candidate solution c is outlined in Algorithm 1. Using this metric, our system optimizes the coherence of the focus structure. As each atomic structure in the ROI will become a focus structure in one of the optimization iteration, the system will eventually display each of them coherently and clearly at least once.

Algorithm 1. Coherence Metric

$f \Leftarrow$ the focus structure
if $V(f) \geqslant t_{sv}$ **then**
  render f alone to get its structure image
  $image_f \Leftarrow$ grayscale image of structure image of f
  $image_c \Leftarrow$ the grayscale image of the rendered image produced by candidate solution c
  $image_{diff} \Leftarrow$ new image for difference storage
  for each pixel (x, y) in $image_{diff}$ **do**
    $image_{diff}(x, y) = |image_f(x, y) - image_c(x, y)|$
  end for
  $P \Leftarrow$ the set of $image_{diff}$ pixels which s occupies
  $H(c) \Leftarrow \sigma^2(P)$
else
  $H(c) \Leftarrow (1 + (t_{sv} - V(f))/t_{sv}) \cdot K_h$
end if

5.3 Context Metric

The relation among structures is very important information. Hence, we want to introduce new information while keeping certain old information in each generated DVRI so that the relation between the old and the new can be easily understood. Our context metric consists of two parts, namely, data-domain context metric and image-domain context metric.

5.3.1 Data-Domain Context Metric

If there is no common structure presented in the generated DVRIs, users may not be able to easily establish relation among the structures revealed in them. Certain shared information should be maintained to serve as critical context. Fig.2 shows an illustration. Some fish organs are shown in the middle of a DVRI (see Fig.2(a)) while some fish bones are shown in another DVRI (see Fig.2(b)). Both features are shown clearly in their own images, however, users have no idea about the relation between the organs and the bones.

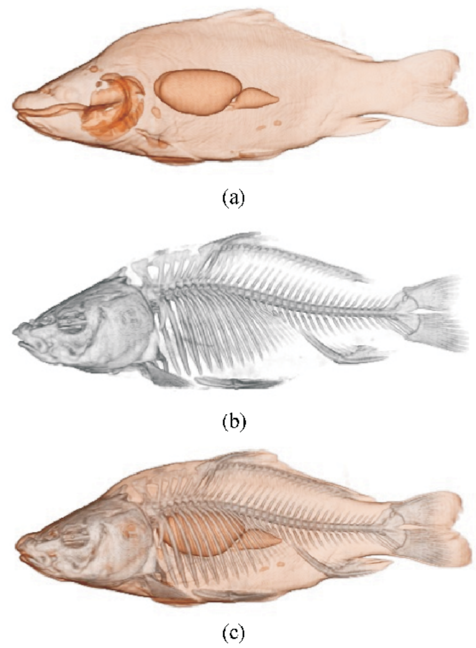


Fig.2. Context problem. (a) Some fish organs are shown in the middle of a DVRI. (b) Fish bones are shown in a DVRI. (c) Both structures, being the context for each other, are shown in the same DVRI. With contextual information, users can easily establish the relation between the structures, i.e., the organs are under the bones.

In the good example (see Fig.2(c)), both structures, being the context for each other, are shown in the same DVRI, so users can easily see that the organs are under the bones. To address this issue, we develop the data-domain context metric to measure how much old information in the data-domain is kept in a generated DVRI to facilitate users' recognition of inter-structure relations.

Given an old DVRI I_{old} and a new DVRI I_{new}, the data-domain context metric $C_{data}(c)$ for I_{new} based on the presented information in I_{old} is defined as

$$C_{data}(I_{new}, I_{old}) = \begin{cases} \dfrac{|S_{new} \cap S_{old}|}{|S_{new}|}, & \text{if } |S_{new}| > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where S_{new} and S_{old} are the sets of presented structures in I_{new} and I_{old} respectively. It measures the percentage of atomic structures previously presented in I_{old} which remain in the new DVRI I_{new}. The higher its value, the more common structures are presented, and therefore the easier the user can relate the new DVRI to the old DVRI using the shared context structures.

5.3.2 Image-Domain Context Metric

As users study the structures by looking at the DVRIs, we need an image-domain metric to estimate the perceived contextual information in addition to the data-domain one. Image similarity measurement is involved in our image-domain context metric. The measurement is based on edge images because of the observation that edges on a 2D image are important perception clues for structures. We adopt the same method for image similarity measurement as in [9] although our use of them is different. To compute the image-domain context metric for I_{new} based on the presented information in I_{old}, we first use a well-established optimal edge detector called Canny edge detector to compute the edge images for I_{new} and I_{old}. Then we apply Gaussian filtering to both edge images before the image comparison. Finally we calculate the percentage of similar pixels of them in the same location and use this percentage as our image-domain context metric $C_{image}(I_{new}, I_{old})$.

While using this image-domain metric alone can sometimes give false results due to inaccurately detected edges on the rendered image, using it together with the data-domain metric can alleviate this problem and give a good estimation.

5.3.3 Context-Aware Exploration Sequence

To have all generated DVRIs introduce new information coherently while keeping a certain amount of old information shared with all the rest of the DVRIs is very difficult. We therefore propose a concept, namely context-aware exploration sequence, which serves as a foundation of the final form of the context metric. A context-aware exploration sequence is a DVRI viewing order in which a DVRI always shares a certain amount of context with its previous DVRI. That is, for a context-aware exploration sequence $\langle x_1, x_2, \ldots, x_n \rangle$, DVRIs x_j and x_{j+1}, with $1 \leqslant j < n$, share a certain amount of context so that the structures shown in them can be easily related with each other. Exploiting this concept, our context metric just needs to ensure that every newly generated DVRI has enough shared context with at least one of the previously generated DVRIs or the original DVRIs. It is because, logically, any newly generated DVRI can be related to an original DVRI in some context-aware exploration sequence if it can be related to any of the previously generated DVRIs.

Given a candidate solution c generating a DVRI I_c and a set of all previously shown DVRIs Θ (i.e., both the previously generated DVRIs and the original DVRIs), the final form of the context metric $C(c)$ of I_c is defined as follows.

$$C(c) = \begin{cases} 0, & \text{if } \exists I_p \in \Theta \text{ such that} \\ & C_{data}(I_c, I_p) \geqslant t_{data} \text{ and} \\ & C_{image}(I_c, I_p) \geqslant t_{image}, \\ (1 + (\delta_{data} + \\ \ \delta_{image}) \cdot K_c, & \text{otherwise,} \end{cases}$$

$$\delta_{data} = \begin{cases} 0, & \text{if } C_{data}(I_c, I_m) \geqslant t_{data}, \\ 1 - \dfrac{C_{data}(I_c, I_m)}{t_{data}}, & \text{otherwise,} \end{cases}$$

$$\delta_{\text{image}} = \begin{cases} 0, & \text{if } C_{image}(I_c, I_m) \\ & \geqslant t_{image}, \\ 1 - \dfrac{C_{image}(I_c, I_m)}{t_{image}}, & \text{otherwise,} \end{cases}$$

where $I_m \in \Theta$ is the DVRI with the minimum value of $(\delta_{data} + \delta_{image})$, i.e., the DVRI with the most contextual information in the set. t_{data} and t_{image} are the threshold values for data-domain context metric and image-domain context metric respectively. If one of those metrics is below its corresponding threshold level, the difference between the calculated value and the threshold will be multiplied by a large constant K_c and reflected in the final context metric value. The larger the context metric value, the worse the context. If both data-domain metric and image-domain metric are satisfactory, 0 will be the resulted value to indicate that the context is good enough.

6 Automatic DVRI Generation

Our system generates DVRIs using genetic algorithm with the quality metrics described in the previous section. Genetic algorithm is a widely used technique for solving optimization problems. The solution encoding design and energy function design are the two major challenges for modeling a problem properly and finding a good solution. Details of genetic algorithm can be found in [7].

6.1 Solution Encoding

One possible encoding scheme is to use a series of numbers to represent the opacity mapping defined by a transfer function. For example, a 1D transfer function can be represented by a 1D array that defines different opacity values for voxels with different intensities. This approach is adopted by He *et al.*[1] and Wu and Qu[9]. This, however, is not flexible enough to solve our problem. In many cases, structures of similar intensities occlude each other. Therefore, not all of them can receive enough visibility when they are always assigned with the same opacity value by the transfer function. With such a limitation, visibility cannot be guaranteed for all the structures no matter how the transfer function is set. The most flexible solution encoding is to use a large set of numbers to define the opacity value for each voxel. With such flexibility, we can precisely control the opacity of each voxel in each DVRI, so visibility can definitely be guaranteed for all voxels. However, the solution space becomes extremely large for the algorithm to find a good solution.

We, therefore, adopt a divide-and-conquer approach to tackle this problem. We first divide the volume into a set of small components called atomic structures. Then for each atomic structure we only need an opacity seed value to automatically generate a transfer function for it. We may proceed to encode a solution with a series of numbers which represent opacity seed values for atomic structures. With this approach, however, atomic structures of the same structures may be assigned with different opacity and thus appear strange and not coherent with each other. Because of this, a structure is divided into atomic structures only when decomposition is necessary to reveal certain part of it. We formulate the solution encoding for the genetic algorithm as an array of floating-point numbers: $(x_1, x_2, \ldots, x_n)$ where x_i is the opacity seed value of structure or atomic structure i. Our solution space is small enough while we can generate high-quality solution that provides visibility guarantee to interested structures.

6.2 Structure Categorization

Each of the structures and atomic structures should be rendered differently as they have different roles and positions in the DVRI, to achieve this, we categorize them according to their roles in the DVRI and their occlusion relations. We first analyze the occlusion relations among the structures and then build a data structure called occlusion graph to store them for efficient use of information later. The occlusion graph is built by having atomic structure represented as nodes and adding directed edges from node u to node v if the structure represented by u occludes that by v. After we have the occlusion graph, we can categorize all the atomic structures as one of the following four categories:

• *Focus Structure.* For each generation, our system selects a focus structure, which is previously not visible or clear enough according to our metrics, to be revealed in the new DVRI. It should become visible enough in the generated DVRI to introduce new information. Therefore their opacity seed values should be set high.

• *Occluding Context.* We should show the focus structure with their context included in the DVRI as well so that they can be related to other structures in the volume for comprehension. The context structures occluding the focus structure are labeled as occluding context. They should be assigned very low opacities, if any, so that they would not occlude the focus structure too much in order to provide visibility guarantees to the voxels of the focus structure. The effect of gradient magnitude opacity-modulation should also be magnified to suppress their uniform regions while enhancing the edges to give just enough clues of their existence and reduce the occlusion to the focus structure. On the other hand, the effect of depth opacity-modulation can be turned off as we do not need to give visibility guarantees to voxels of occluding context.

• *Occluded Context.* Context structures which are occluded by focus structure are included in the occluded context category. Similar to the reasoning for occluding context, the effect of depth opacity-modulation can be turned off. The effect of gradient magnitude opacity-modulation can also be turned off as the structures in this category are seriously occluded by the focus structure. We do not need any suppression or enhancement. As they are occluded by the focus structure, to preserve their existence as a context, we assign high opacity values to them to allow them to be still visible.

• *Other Context.* Any other structures not included in the above three categories are categorized as other context. They are neither occluding nor occluded by the focus structure. We can directly give them medium opacity values to serve as surrounding context. Opacity-modulation can simply be turned off for them.

6.3 Energy Function and Solution Search

The energy function we use consists of the three quality metrics described in Section 5. Note that as the visibility metric is combined into the coherence metric, we only have two terms. The energy function for a candidate solution c is defined as $E(c) = H(c) + C(c)$. The optimization process will not end successfully if $E(c)$ is greater than or equal to one of the large constants K_h and K_c, that is, either one or both of the visibility guarantee and context constraint is not satisfied. On the other hand, if both of the visibility guarantee and context constraint are satisfied, the optimization process will optimize the solution for the best information coherence in the given time. We run the genetic algorithm for a certain number of iterations then its best candidate is checked for its quality. If the candidate can introduce new information while it can be related to the previously generated DVRIs, then it will be outputted.

Otherwise, we continue to run the genetic algorithm for the same focus structure. The whole process is repeated until all the interesting atomic structures have been revealed in high quality at least once.

7 Animation Generation

A subset of the generated DVRIs can be chosen by the users to generate an animation that presents the missing information comprehensively. The selected DVRIs will serve as keyframes for the animation. Transitional DVRIs will be generated between the keyframes to provide smooth transition. They are generated by linearly interpolating the opacity value of each voxel from one keyframe DVRI to another.

7.1 Keyframe Ordering

While users can define the keyframe ordering themselves, our system also provides an automatic keyframe ordering approach. As the most important thing in a keyframe DVRI is the focus structure, logically the ordering should be based on it. A natural ordering of the keyframes is the depth ordering of the focus structures presented in them. That is, the keyframes should be ordered in a way such that the volume data is explored from outside to the inside and from the front to the back. To sort the focus structures, if two focus structures are of different classes according to the provided segmentation, they will be ascendingly sorted according to the minimum depths (shortest distances to the viewplane) of the segments they belong to. Otherwise, they will be ascendingly sorted according to their own minimum depths.

7.2 Temporal Coherence

After the keyframe ordering is obtained, we need to fix a temporal coherence issue that possibly exists between consecutive keyframes. The problem is illustrated in Fig.3. Suppose the keyframe ordering is Fig.3(a), then Fig.3(b), and then Fig.3(c), it can be observed that the engine body in the middle keyframe Fig.3(b) has different opacity compared with Figs. 3(a) and 3(c). In effect, the animation will show a highly transparent engine body, then a less transparent engine body, and finally a highly transparent engine body again. As engine is a contextual structure in all the keyframes, the change of opacity of unnecessary. Moreover, this kind of opacity fluctuation during the animation can introduce inconsistence and confusion issues. Therefore, it is desirable to have those kinds of contextual structures have stable opacity assignments throughout the whole animation.

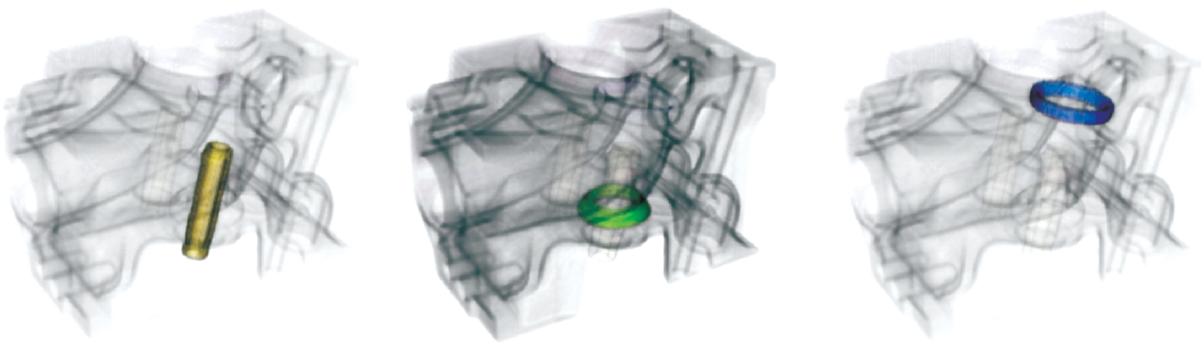
To address this problem, for each structure, our system first analyzes all the keyframes, and records the lowest non-zero opacity assigned when the structure is used as context. Then each keyframe will be checked to make sure the context constraint will not be violated, according to the context metric, after the lowest opacity in record is assigned to the structure. If the context condition is still good for all keyframes after the change, then the change will be actually committed. This procedure can standardize the opacity of the contextual structures when it is possible.

8 Experimental Results

Experiments are all performed on a standard PC machine (Intel Core 2 Duo E6400, 3 GB RAM) equipped with a GeForce 8600GT graphics card.

8.1 CT Head

An experiment is conducted on a CT head dataset ($128\times128\times231$). The brain is selected as the ROI from the segment gallery and a DVRI, as shown in Fig.4(a), is rendered. Four images are then generated by our system in 157 seconds to guarantee the whole brain structure to have at least 80% visibility. Fig.4(b) shows the



(a) (b) (c)

Fig.3. Temporal coherence problem. (a)$\sim$(c) Keyframes in a user-defined order. The engine body has different opacity in (b) compared with (a) and (c). This introduces inconsistence and confusion in the animation.

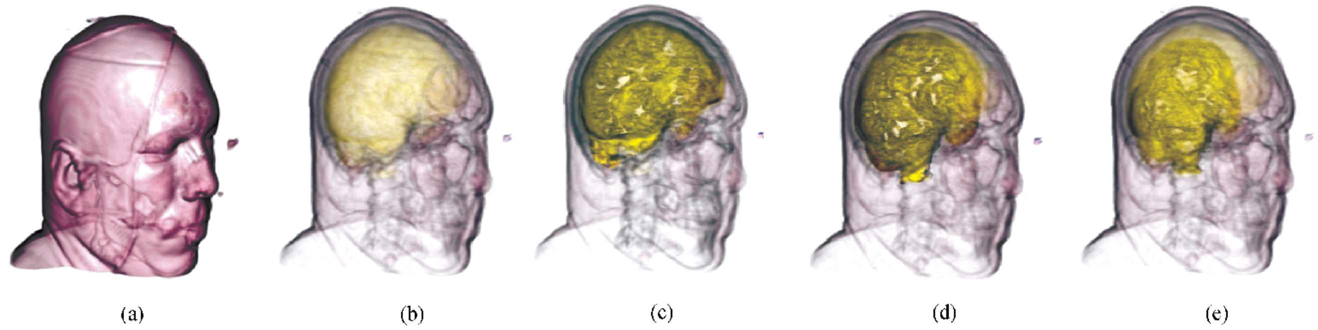


Fig.4. CT head experiment. (a) User-generated image. (b) System-generated DVRI that shows the outer view of the brain. (c)∼(e) Other system-generated DVRIs that show the inner view of the brain.

outside view of the brain structure which is in yellow, and then the inner view of the brain structure is revealed in Figs. 4(c), 4(d), and 4(e). The skin and the skull shown in the original DVRI are preserved as context. The generated DVRIs help the user to explore the brain structure almost completely. The amount of context is optimized in the optimization process so that the focus structures (i.e., the atomic structures of the brain) can be shown clearly while the context is still visible to provide important contextual information. An animation is then produced to provide a comprehensive and smooth presentation. The animation, which is generated in 135 seconds, shows what is behind the skin and inside the brain gradually with well-preserved context.

Note that as the head dataset is a CT dataset, the quality of the brain structure is not very good. However, this experiment demonstrates how our system can be used to provide visibility guarantee to interesting structures. This can help the user to explore those structures thoroughly.

8.2 CT Engine

We conducted another experiment on a CT engine dataset ($256 \times 256 \times 128$). The user-rendered DVRI (Fig.5(a)) shows some interesting structures in different colors but most of them are not very clear. The user wants to see them clearly and check if there is anything missing in the nearby region. The ROI is specified by the user with brushing which is indicated by aqua color in Fig.5(b). Since the engine structures are well segmented and do not have anything interesting inside them, the structure visibility guarantee threshold is set to 50% to avoid unnecessary structure decomposition that is used to reveal the inner parts of the structures. Our system automatically detects missing or unclear structures and generates a set of DVRIs to reveal them. In this experiment, the system generates two DVRIs to reveal all the missing or unclear structures within the ROI. The generation process takes 53 seconds. Fig.5(c) shows clearly almost all the interesting structures which are in different colors, however, part of the red structure is occluded and thus missing. Therefore, another generated DVRI, as shown in Fig.5(d), reveals the red structure completely by lowering the opacity of the occluding parts (i.e., the green structure and the yellow structure). In both generated DVRIs, the engine body is preserved as the context. Finally, an animation based on the original DVRIs and the two generated DVRIs is generated in 73 seconds.

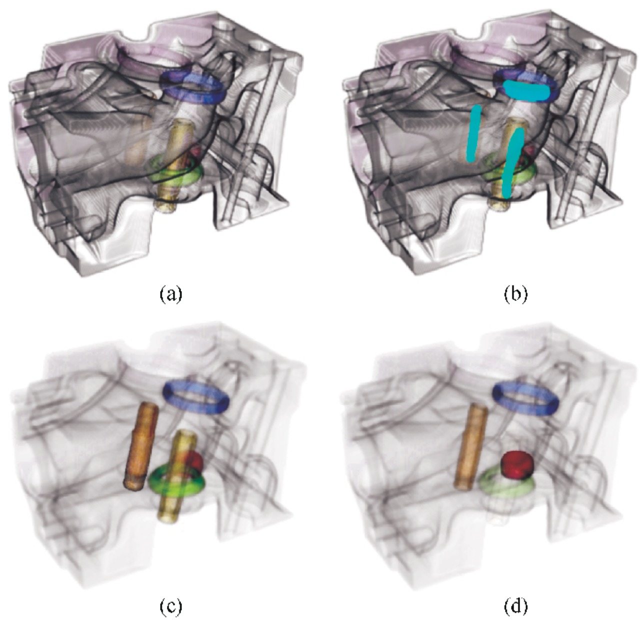


Fig.5. CT engine experiment. (a) User-generated image. (b) Same user-generated image with brushing ROI selection on it. Strokes are in aqua color. (c) System-generated DVRI showing all the interesting structures indicated by the user. (d) Another system-generated DVRI clearly revealing the red structure that is occluded before.

9 Discussion

Viewing the slices of a volumetric dataset one by one can also guarantee the visibility of all voxels. However, the coherence of structure is destroyed as a 3D structure becomes a stack of 2D slices. Moreover, the spatial inter-structure relation shown by 2D slices is not as clear as that in a 3D overview given by a DVRI. Our approach is based on DVR that shows important information that can only be effectively conveyed in 3D. In addition, we guarantee visibility, preserve coherence, and provide critical context.

To reveal all the features in a volumetric dataset using direct volume rendering is a challenging task. One major problem is that users are not informed whether all the information has been revealed. Even if they know what is missing or unclear, tuning a transfer function to reveal those information still requires their tremendous effort. On the other hand, the missing information can be guaranteed to be presented in a coherent way with the automatic approach, which requires less effort from users, in our framework as demonstrated in this chapter.

Ghosted view is a classical approach in volume exploration that can improve the visibility of an important structure. It has many variances but the basic approach is the same. Given a structure of interest, the ghosted view removes the occluding region to reveal the structure. Nonetheless, it does not guarantee visibility of the whole structure that there could be important region within the structure of interest left unrevealed due to self-occlusion. With our approach, self-occlusion is not a problem as we guarantee visibility of voxels, thus the occluded region will be shown in one of the generated DVRIs.

Removing the visible voxels from front to back to show occluded regions also has a major drawback that it may not provide critical context. To show the structures of interest, such a context may be removed as well in the process. For example, the skin of a human head would have to be completely removed before the brain can be seen. Our approach is flexible to provide contextual information of different depths and remove only minimal amount of the occluding structures that is necessary to show the occluded structures clearly while the occluding structures can still be served as important context.

The applications of our method are twofold: as a safety check and as a volume exploration method. In one scenario, users can use other volume visualization methods to explore the data and save some representative DVRIs. After they finish the visualization process and have a good understanding of the data, they can use our system to check if any important information is missing. Thus, our method serves as a safety check tool which can be incorporated into existing volume visualization systems. In another scenario, users can first generate some initial DVRIs using some automatic or semi-automatic methods. Then starting from these initial DVRIs, our system can be employed to semi-automatically generate new DVRIs one by one. Because each new DVRI is guaranteed to contain some new information and also keep some old information, users can use our method to explore different regions of the data. In this manner, our method can serve as a volume exploration scheme.

Though the experiments are promising, our current metrics are still primitive and user studies are needed to validate them. For example, our energy function includes measurement of coherence which is a complicated issue involving human perception. All the generated DVRIs have the same viewpoint and lighting parameters. This is because we need to limit the degrees of freedom as the current problem with fixed viewpoint is already highly challenging and complicated. Some speedups exploiting this setting also become possible.

10 Conclusion and Future Work

In this paper, we have developed a framework to semi-automatically generate DVRIs and an animation for a given volumetric dataset and existing DVRIs. The generated DVRIs can reveal all the missing information in users' region of interest, and thus can improve the safety of a volume visualization system and boost the confidence of users in DVR. It promotes the wider adoption of DVR in real applications. Our framework can guarantee that the missing information will be presented in a coherent way while some old information from the previous DVRI will be kept to provide users some familiar context to understand relations among structures. A recursive procedure is used to generate DVRIs using genetic algorithm which optimizes a carefully-designed energy function consisting of visibility, coherence, and context terms. Our method can be used as a safety check for volume visualization systems as well as a volume exploration method which allows users to start from some initial DVRIs and gradually and recursively generate additional DVRIs until all interesting information is revealed.

Our framework can be extended in several ways. Currently, all the DVRIs are generated from the same viewpoint under the same lighting setting. In the future, we will investigate how to generate a set of DVRIs with an additional freedom of changing the lighting and viewpoint. We believe there are many possibilities to measure the quality of a DVRI. We plan to incorporate more metrics such as contrast and color harmonization into our extensible framework. Moreover, we will exploit GPU for system acceleration. We will also apply our method as a built-in feature to some real systems.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] He T, Hong L, Kaufman A, Pfister H. Generation of transfer functions with stochastic search techniques. In *Proc. IEEE Visual. Conf.*, San Francisco, USA, Oct. 27-Nov. 1, 1996, pp.227-234.

[2] Kindlmann G, Durkin J W. Semi-automatic generation of transfer functions for direct volume rendering. In *Proc. IEEE Symposium on Volume Visualization*, Research Triangle Park, USA, Oct. 24, 1998, pp.79-86.

[3] Kniss J, McCormick P, McPherson A, Ahrens J, Painter J, Keahey A, Hansen C. Interactive texture-based volume rendering for large data sets. *IEEE Computer Graphics and Applications*, 2001, 21(4): 52-61.

[4] König A, Gröller E. Mastering transfer function specification by using volumepro technology. In *Proc. Spring Conference on Computer Graphics*, Budmerice, Slovakia, Apr. 25-28, 2001, pp.279-286.

[5] Tzeng F Y, Lum E B, Ma K L. A novel interface for higher-dimensional classification of volume data. In *Proc. IEEE Visual. Conf.*, Seattle, USA, Oct. 19-24, 2003, pp.505-512.

[6] Correa C, Ma K L. Visibility-driven transfer functions. In *Proc. IEEE Pacific Visualization Symposium*, Beijing, China, Apr. 20-23, 2009, pp.177-184.

[7] Mitchell M. An Introduction to Genetic Algorithms. Cambridge: MIT Press, MA, USA, 1996.

[8] House D, Bair A, Ware C. On the optimization of visualizations of complex phenomena. In *Proc. IEEE Visualization Conf.*, Minneapolis, USA, Oct. 23-28, 2005, pp.87-94.

[9] Wu Y, Qu H. Interactive transfer function design based on editing direct volume rendered images. *IEEE Trans. Visual. Comput. Graph.*, 2007, 13(5): 1027-1040.

[10] Sheikh H R, Sabir M F, Bovik A C. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Processing*, 2006, 15(11): 3441-3552.

[11] Sheikh H R, Bovik A C, Cormack L. No-reference quality assessment using natural scene statistics: JPEG2000. *IEEE Transactions on Image Processing*, 2005a, 14(11): 1918-1927.

[12] Gunawan I P, Ghanbari M. Reduced-reference video quality assessment using discriminative local harmonic strength with motion consideration. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008, 18(1): 71-83.

[13] Wang Z, Bovik A C. Modern Image Quality Assessment. Morgan and Claypool Publichers, 2006.

[14] Wang C, Garcia A, Shen H W. Interactive level-of-detail selection using image-based quality metric for large volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2007, 13(1): 122-134.

[15] Wang C, Ma K L. A statistical approach to volume data quality assessment. *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14(3): 590-602.

[16] Gröller M E, Purgathofer W. Coherence in computer graphics. Technical Report, TR-186-2-95-04, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, March 1995, Contact: technical-report@cg.tuwien.ac.at.

[17] Lundstrom C, Ljung P, Ynnerman A. Local histograms for design of transfer functions in direct volume rendering. *IEEE Trans. Visual. Comput. Graph.*, 2006, 12(6): 1570-1579.

[18] Weiskopf D, Engel K, Ertl T. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Trans. Visual. Comput. Graph.*, 2003, 9(3): 298-312.

[19] Viola I, Kanitsar A, Gröller M E. Importance-driven feature enhancement in volume visualization. *IEEE Trans. Visual. Comput. Graph.*, 2005, 11(4): 408-418.

[20] Krüger J, Schneider J, Westermann R. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Trans. Visual. Comput. Graph.*, 2006, 12(5): 941-948.

[21] Bruckner S, Grimm S, Kanitsar A, Gröller M E. Illustrative context-preserving exploration of volume data. *IEEE Trans. Visual. Comput. Graph.*, 2006c, 12(6): 1559-1569.

[22] Rezk-Salama C, Kolb A. Opacity peeling for direct volume rendering. *Computer Graphics Forum*, 2006, 25(3): 597-606.

Wai-Ho Mak received the B.Eng. degree (first-class honors) in computer science and the M.Phil. degree in computer science and engineering both from the Hong Kong University of Science and Technology (HKUST) in 2007 and 2009 respectively. His research interests include scientific visualization and information visualization.

Yingcai Wu is a post-doctoral researcher in the Department of Computer Science at The University of California, Davis. He obtained his B.Eng. degree (2004) in computer science and technology from South China University of Technology, China, and his Ph.D. (2009) degree in computer science and engineering from the HKUST. His research interests are in medical volume visualization and visual analytics.

Ming-Yuen Chan received the B.Eng. degree (2003) in computer engineering from the University of Hong Kong and the M.Phil. (2006) and the Ph.D. degrees (2009) in computer science from the HKUST. His research interests include scientific visualization and medical imaging.

Huamin Qu is an associate professor in the Department of Computer Science and Engineering at the HKUST. His main research interests are in visualization and computer graphics. He has conducted a wide range of research on scientific visualization, information visualization, real time graphics systems, virtual reality, and medical imaging. He received the B.S. degree in mathematics from Xi'an Jiaotong University, China, the M.S. and the Ph.D. degrees in computer science from the Stony Brook University, New York.