Palette-Style Volume Visualization

Yingcai Wu[1], Anbang Xu[1], Ming-Yuen Chan[1], Huamin Qu[1], and Ping Guo[2]

[1]The Hong Kong University of Science and Technology
[2]Beijing Normal University

Abstract

In this paper we propose a palette-style volume visualization interface which aims at providing users with an intuitive volume exploration tool. Our system is inspired by the widely used wheel-style color palette. The system initially creates a set of direct volume rendered images (DVRIs) manually or automatically, and arranges them over a circle in 2D image space. Based on the initial set of DVRIs called primary DVRIs which imitate the primary colors in the color wheel, users can create more DVRIs on the wheel using PhotoShop-style image editing operations such as the fusing operation. With our system, non-expert users can easily navigate and explore volumetric data. In addition, users can always know where they have been, where they are, and where they could go in a visualization process and hence redundant exploration can be avoided.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Graphics Systems

1. Introduction

Direct volume rendering (DVR) is a powerful tool for users to analyze volumetric data. It generates *Direct Volume Rendered Images* (DVRIs) of the data without any intermediate geometry by a *transfer function* (TF) which maps data properties like intensity and gradient to optical properties such as color and transparency. DVR is a powerful and flexible volume visualization technique and can help users gain deep insight into the data. However, nonintuitive volume visualization interfaces in current systems prevent its wide adoption in practice. It is very challenging for non-expert users to use DVR because usually many parameters (e.g., TF, viewpoint, lighting) need to be fine tuned to generate an informative DVRI.

In recent years, researchers have become increasingly interested in intuitive volume visualization interface design [KD98, KKH02, KG01, TLM05]. Most previous approaches focus on providing users with efficient transfer function design interfaces. With these approaches, transfer functions and their corresponding DVRIs can be generated automatically, semi-automatically, or manually. Even for very complicated datasets, many intermediate DVRIs can be generated in a short time during a visualization process with more and more powerful computers. These intermediate DVRIs may contain some useful information about the data and should not be discarded. Otherwise, users may not know
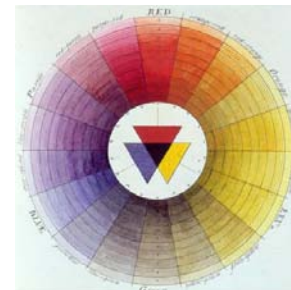
Figure 1: *Moses Harris: the first color wheel to classify red, blue and yellow as the three primary colors .*

whether they have explored certain features with some rendering parameters (e.g., camera and lighting settings). Hence the method to organize these intermediate DVRIs intuitively and efficiently has become an important research issue. First, users may get lost in a long visualization process and, therefore, may need to backtrack to some cached DVRIs. An intuitive interface is thus needed to facilitate users to find some specific DVRIs. Second, users may want to explore new features and generate new DVRIs based on the rendered ones. Therefore, an interface that allows users to easily generate new DVRIs from a set of cached DVRIs should be provided.

In this paper, we develop a novel volume visualization interface which aims at increasing exploration intuitiveness, reducing exploration redundancy, and saving and sharing exploration results. This approach is motivated by a traditional color specification tool - the circular color palette shown in Fig. 1. The palette usually contains several primary colors which lie at the center or on the edge of the palette. More colors can be created by mixing the primary colors and the newly created colors will lie in the palette accordingly. With this color palette, users can readily select or specify a desired color.

Based on this observation, a palette-style interface is proposed for intuitive DVRI generation. Initially, the system generates a number of opacity TFs and their DVRIs called primary opacity TFs and DVRIs like the primary colors in the circular color palette (or color wheel) semi-automatically [KD98] or automatically [MAB*97], and arranges them near the edge of a DVRI wheel. More DVRIs of different opacity TFs can be created by mixing the primary DVRIs in a similar way of mixing primary colors in the circular color palette. The system enables users to further explore the data using the created opacity TF with different color TFs, viewpoints, and/or lighting parameters in a separate radial graph. The radial graph can help users arrange DVRIs automatically so that better DVRIs lie closer to the center. Our system can also be used to store the rendered DVRIs based on their similarity. Some primary DVRIs that represent distinct features in the data are first selected automatically or manually from the rendered DVRIs. These primary DVRIs will be positioned along the boundary of the DVRI wheel. Our system will then compute the similarity between all other DVRIs and the primary DVRIs and position them accordingly in the wheel so that similar DVRIs will be grouped together.

This paper is organized as follows: After reviewing previous work in Section 2, we briefly introduce our system in Section 3. We then present the palette-style DVRI wheel and its supported operations in Section 4. A radial graph for detailed exploration of a volume using a specific opacity TF is described in Section 5. After that, experiment results and discussions are presented in Section 6. We draw conclusions and suggest some future work in Section 7.

2. Related Work

User Interface is critically important for successful volume visualization, as it serves as a bridge for communication between the users and the system and strongly influences users' understanding of the data [TM05]. To "keep users informed about what is going on" is an important criterion for successful user interface [NM90]. To address this issue, researchers have proposed a variety of user interfaces [Ma99, JKM01, KNS04, TPM05, JKMG07] by providing users helpful history mechanisms. Ma [Ma99] developed a classical user interface called image graphs for the visual representation of the data exploration process. His approach

can effectively show end-users the relationship between parameter changes and the corresponding resulting images. Moreover, it allows users to operate on the nodes and edges of the recorded graph dynamically and supports information reuse and sharing. Jankun-Kelly and Ma [JKM01] proposed a spreadsheet-like interface for visualization exploration and encapsulation. Their method provides users with an efficient way for data examination and result comparison, and helps users reduce exploration redundancy and facilitate collaboration. Tory *et al.* [TPM05] introduced a new volume visualization interface based on parallel coordinates. Their system gives users a comprehensive view of different rendering options and an intuitive history mechanism which helps users readily backtrack to previous visualization settings. Kreuseler *et al.* [KNS04] developed a tree structure style history mechanism which helps users keep track of the whole data exploration process and supports redo and undo operations on the history tree. VisTrails presented by Bavoil *et al.* is an interactive visualization system with multiple views. It utilizes the spreadsheet [JKM01] for multiple-view visualizations and maintains the derivation of a large number of visualization results. Jankun-Kelly *et al.* [JKMG07] proposed a framework based on a novel P-Set model for users to encapsulate, share, and evaluate the whole visualization process. Compared with these systems, our interface focuses on organizing DVRIs in an intuitive manner and generating new DVRIs based on some key DVRIs.

TF design is another research hotspot of volume visualization and numerous approaches have been proposed in the past [PLB*01, Kin02]. Marks *et al.* [MAB*97] developed an image-centric TF design method which creates many perceptually different DVRIs automatically and arranges them in 3D space for users to select based on input parameters. Compared with their method, our approach organizes DVRIs based on their similarity instead of input parameters and allows users to use a palette-style interface for creating opacity TFs semi-automatically. König and Gröller [KG01] proposed an intuitive user interface which provides users with several TF specification tools for each search domain. Kindlmann and Durkin [KD98] presented a classic data-centric TF design method that can easily obtain good TFs semiautomatically. Their method can be integrated in our system for creating primary DVRIs. Kniss *et al.* [KKH02] extended Kindlmann and Durkin's approach with several useful TF widgets and a novel dual-domain operation. However, users usually cannot distinguish distinct features using the existing approaches [KD98, KKH02] based on the traditional histograms. To address this issue, Šereda *et al.* [vBSG06] presented a TF design method based on the novel LH histogram. An interactive multi-scale TF design method [LSM06] based on filter banks has recently been proposed. It is particularly effective for classifying fuzzy or noisy features. Our method is based on an intuitive TF design method [WQ07] which provides users with fusing, blending, and deleting operations on DVRIs for TF specification. Another high-level TF de-

(a)   (b)
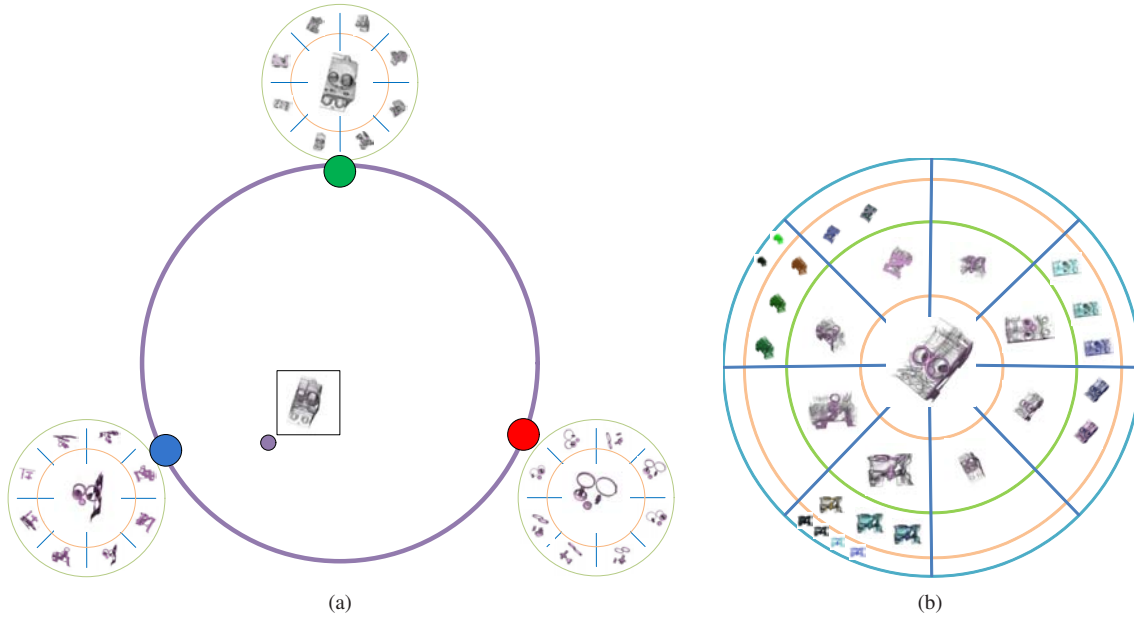
Figure 2: *(a) A palette-style DVRI wheel; (b) A radial graph for detailed exploration.*

sign approach described by Salama *et al.* [SKK06] can also be integrated into our system.

Radial Graph Visualization is an important topic in information visualization and many different approaches have been proposed. Eades [Ead92] first introduced radial graph layouts that are widely used in graph-based information visualization [TBET98]. MoireGraphs [JKM03] developed by Jankun-Kelly and Ma is a focus + context visualization method for graphs with image nodes. In this paper, we utilize a radial layout which is similar to MoireGraphs to organize the created DVRIs. However, we use a new metric based on image quality to compute the distance of DVRIs to the center, so that the DVRIs with higher image quality lie closer to the center and have larger display sizes. Moreover, our system divides the radial graph into different sectors automatically in terms of viewpoint, and so DVRIs with similar viewpoints can be grouped together.

3. Interface Overview

Our system consists of two major components, *i.e.*, a palette-style DVRI wheel for creating opacity TFs and their DVRIs from the primary opacity TFs (see Fig. 2(a)), and a radial graph for detailed exploration of a specific opacity TF (see Fig. 2(b)). In the beginning, an initial set of opacity TFs and their DVRIs, *i.e.,*, primary TFs and DVRIs, are created manually or automatically using existing approaches [KD98, MAB*97]. The primary DVRIs will be positioned on the boundary of the DVRI wheel evenly. Users can then manually adjust their positions by drag-and-drop. Similar to

the circular color palette, the system can synthesize an opacity TF and create its DVRI accordingly at any position in the palette using the fusing operation [WQ07].

Users can further explore a DVRI selected from the DVRI wheel using different colors, lighting parameters, and viewpoints in a radial graph. Fig. 2(b) shows a fully deployed radial graph. The radial graph places the focus DVRI at the center surrounded by the context DVRIs. Compared with MoireGraphs [JKM03], the radial graph used in our system can automatically divide the DVRIs into multiple sectors in terms of viewpoint, and then arranges the DVRIs automatically according to their image quality or other user-chosen criterion so that better DVRIs with high image quality will lie closer to the center.

4. Palette-style Volume Exploration and DVRI Organization

4.1. Palette-Style DVRI Wheel

The basic idea of the palette-style DVRI wheel interface is to help users design opacity TFs intuitively and this is a similar way to mix primary colors in the circular color palette as shown in Fig. 1. Since we use DVRIs to represent their opacity TFs in the interface, this interface is simply called DVRI wheel. In the traditional circular color palette, primary colors should be defined first and a complete color wheel is then constructed by mixing the primary colors. Similarly, users first need to generate a set of primary DVRIs that usually represent distinct features in the volumetric data for the

DVRI wheel and then other DVRIs can be intuitively generated by mixing primary DVRIs. In a color wheel, users can choose their desired colors directly from the wheel by selecting a point on the wheel. Intuitively, the closer a selected point is to a primary color, the more similar the generated color is to the primary color. This inverse proportion property is held in our palette-style DVRI wheel for creating more opacity TFs and their DVRIs as well.

4.2. Primary DVRI Generation

Primary DVRIs can be created by experts manually or semi-automatically [KD98] which makes boundaries of structures in volumetric data visible. The semi-automatic approach creates TFs from a special histogram volume structure which captures the relationship between the data value and its first and second directional derivatives. Non-expert users may find it not easy to use as they still have to work in a TF domain. However, this method is appropriate for experts to create primary opacity TFs and their DVRIs. For non-expert users, other methods like [SKK06] can be adopted for primary DVRI generation without the knowledge of TFs.

4.3. Editing Operations on The DVRI Wheel

After creating primary DVRIs, our system allows users to generate more DVRIs from the primary ones intuitively. Users just need to select a point on the DVRI wheel and indicate how to mix the DVRIs (actually their opacity TFs). To create expected DVRIs from primary DVRIs, a high-level TF design method [WQ07] based on image similarity is employed. This approach helps users, both intuitively and interactively, design a TF for fusing multiple features in distinct DVRIs into a comprehensive one. Each operation is transformed into an optimization problem with an energy function which is based on the image similarity values between the expected DVRI and the primary DVRIs. By minimizing the energy function, a desired TF for creating the expected DVRIs can be obtained. The system exploits a genetic algorithm [Mit96] to solve the optimization problem since it can find sufficiently good solutions in most cases by evolving automatically in the TF domain. These PhotoShop-style editing operations are easy to use and can be incorporated seamlessly into our palette-style DVRI wheel. Our interface can use these editing operations to create smooth animations for focus + context visualization as well.

4.3.1. Fusing Operation

Our palette-style interface naturally supports the fusing operation just like mixing colors on the color wheel. Users only need to select a point P in the palette to indicate the contributions (*i.e.*, the expected similarity value V_k in Eq. 1) of each primary DVRI to the expected DVRI. The closer P is to a primary DVRI, the more similar they are, *i.e.*, V_k is inversely proportional to d_i which is the distance from the user selected point P_k to the primary DVRI i as shown in Fig. 3(a).
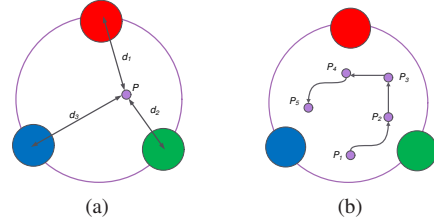

Figure 3: *(a) Fusing operation; (b) Animation generation.*

The fusing operation creates a new comprehensive DVRI by combining the primary DVRIs. The energy function used in the fusing operation is defined as follows:

$$E = \sum_{k=1}^{n} V_k * |V_k - S_k| \qquad (1)$$

where S_k is the similarity value obtained by a contour-based similarity method, and V_k represents the similarity value expected by users between the expected DVRI and the primary DVRI k. By minimizing E, we can obtain a desired comprehensive DVRI (details can be found in [WQ07]).

4.4. Animation Generation

Animation generation is an important application of our interface. This can be used to reveal 3D relationships between the different structures more effectively than a still image [WQ07]. Users can create an animation for volume visualization using the DVRI wheel intuitively based on the fusing operation. They just need to select a series of points, *e.g.*, $P_1, \ldots, P_5$ in Fig.3(b), on the DVRI wheel in sequence. These selected sequential points constitute a path where a number of DVRIs can be created by the fusing operations accordingly (see Section 4.3.1). A smooth animation can be created with these DVRIs for focus + context volume visualization [WQ07].

4.5. Support for Multiple Viewpoints

In our system, primary DVRIs can be generated from different viewpoints. However, image similarity can be computed only for DVRIs rendered from the same viewpoint. To solve this problem, we automatically re-render the primary DVRIs from all available viewpoints that generate these DVRIs and arrange them in a radial graph shown in Fig.2(a). The radial graph will be described in more details in Section 5. After the user clicks on a point in the DVRI wheel, a set of DVRIs from different viewpoints will be generated by mixing primary DVRIs. The user may then choose to cache one or several DVRIs in the DVRI wheel.

4.6. DVRI Organization

Our system can also be used to organize user-rendered DVRIs into a DVRI wheel. As we mentioned before, many

intermediate DVRIs may be generated automatically or semi-automatically for a complicated dataset. Organizing these DVRIs in an intuitive manner may greatly facilitate users to explore data. To organize DVRIs into a DVRI wheel, some primary DVRIs need to be chosen and positioned on the boundary of the DVRI wheel by the users manually. These primary DVRIs usually represent some distinct features in the data. For every other DVRI, our system will automatically compute the similarity of this DVRI and every primary DVRI and then use the similarity value as a normalized distance between the DVRI and the primary DVRI in the DVRI wheel. Suppose there are N primary DVRIs and their positions in the DVRI wheel are $P_1, P_2,..., P_N$, respectively. If the similarity values of a DVRI and primary DVRIs are $S_1, S_2,..., S_N$, respectively, the final position of this DVRI in the DVRI wheel can be computed as follows:

$$P = \frac{\sum_{k=1}^{N} S_k \times P_k}{\sum_{k=1}^{N} S_k} \qquad (2)$$

By using this scheme, similar DVRIs will be positioned close to one another. Users can intuitively find a cached DVRI just like searching for a color in a color wheel.

5. Radial Graph for Detailed Exploration

After generating an opacity TF and its DVRI in the DVRI wheel, users may need to further explore the volume using the opacity TF with different lighting parameters, color TFs, and viewpoints. To facilitate this task, we propose a new radial graph style interface that arranges the resulting images based on viewpoints and image quality. This radial graph helps users organize the previously-created DVRIs and keep them informed on what is going on to reduce exploration redundancy. Moreover, with the graph layouts, the results can be readily shared with others [JKMG07].

5.1. Graph Layout

Fig. 4 shows the radial graph layout which consists of multiple concentric circles. Suppose the radii for these concentric circles are $r_1, \ldots, r_n$ from inside to outside and defined as:

$$r_i = r_{i-1} + \frac{(r_{i-1} - r_{i-2})}{2} \qquad (3)$$

where $i \geq 3$ and $r_2 = 2r_1$, and $r_1 = C$ where C is determined by users. The graph will then put the DVRIs into these concentric circles. The closer a DVRI is to the center, the larger size it has. The center of the graph is a real 3D volume that is currently being explored and encircled by previously-created DVRIs. Users can change the rendering parameters such as color TFs and lighting coefficients to explore the volume thoroughly and create more DVRIs.

In a visualization process, newly-created DVRIs can be assigned to different sectors according to their rendering
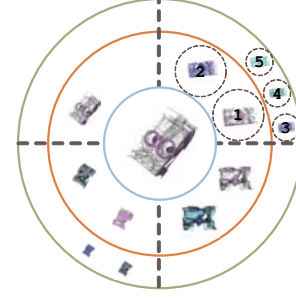
Figure 4: *A radial graph for detailed volume exploration with the same opacity TF. The graph is divided into multiple sectors for storing the DVRIs created from different viewpoints. In each sector, the DVRIs are sorted in terms of image quality so that better DVRIs have larger sizes and are closer to the center.*

viewpoints. Each sector contains all DVRIs rendered from the same viewpoint. The sectors can either be determined by the system automatically or indicated by users. When a new DVRI is created with a new rendering viewpoint, our system will automatically insert a new sector into the radial graph and rearrange the layout.

After the sector is determined, the new DVRI is inserted into the sector and its distance to the center is determined according to its image quality (see Section 5.2). The DVRI with high image quality will lie closer to the center and, therefore, has a larger display size. This emphasizes the focus (DVRIs with higher image quality) while subjugating the context (DVRIs with lower image quality). As we employ a red-black tree structure to store the DVRIs for each sector in memory, the time complexity of inserting a DVRI to a sector is $T + O(\log n)$, where T is the time spent on image quality evaluation and n is the number of DVRIs in the current sector. The DVRIs are kept sorted in ascending order in terms of their image quality in the tree after inserting a new DVRI, and, therefore, we do not need to perform any sorting operation when drawing the sector. Thus, the time complexity for drawing is $O(n)$ for traversing the tree.

5.2. Image Quality Evaluation

To arrange the DVRIs according to image quality, an image quality evaluation scheme needs to be established. As it is subjective to judge the quality of an image and the quality depends on the specific visualization purposes, we adopt the contrast as a measure for general purposes. It is an objective metric to estimate visual variations in images. The features and details in an image with a higher contrast are always better perceived by viewers. To estimate the contrast of an image, we follow the method in [CXS03] where contrast is interpreted as the standard deviation of the pixel values in

the image:

$$\sigma = \sqrt{\frac{1}{|\Omega|}\sum_{\Omega}(v(i)-\mu)^2} \qquad (4)$$

where Ω is the image and $v(i)$ is the intensity of pixel i, and μ is the mean value of all pixels in the image.

This statistical measure can indicate the variation of pixel value in an image. A homogenous image with little visually recognizable features would result in a low standard deviation. This standard deviation value σ also indicates the utilization of colors and provides useful information for the graph layout so that better images can be obtained.

6. Experiment Results and Discussions

Our system was implemented in C++ based on the VTK 5.02 library [Kit] and was tested on a Pentium(R) 4 3.2GHz PC with 2GB RAM and an NVIDIA Geforce 8800 GTX GPU with 768MB RAM. In the following experiments, users utilized our system to explore several volumetric datasets. Since our system is very intuitive and similar to the traditional circular color palette, common users can readily explore and analyze the data with the interface.

6.1. Volume Exploration

To demonstrate the intuitiveness of the palette-style DVRI wheel for volume exploration, we carried out three experiments on different datasets including a CT carp dataset ($256 \times 256 \times 512$), a CT engine dataset ($256 \times 256 \times 128$), and an MRI human head dataset ($256 \times 256 \times 256$). Before starting these experiments, just like specifying the primary colors on the color wheel, the primary DVRIs were specified either manually or semi-automatically [KD98]. After that, a DVRI wheel is presented to users with these primary DVRIs. Each primary DVRI reveals one distinct feature in the data. In the experiments, the users selected an arbitrary point on the DVRI wheel and the system generated a DVRI from the primary DVRIs based on the point position.

We carried out the first two experiments to demonstrate the effectiveness of our interface for providing users with the fusing operation. Fig. 5(a) and 5(b) show two DVRI wheels that were captured when they were being used for the operation. Three small radial graphs were placed around each DVRI wheel. Each graph contains multiple DVRIs generated from different viewpoints for a distinct feature. Users compared the viewpoints in the radial graphs side by side and selected an appropriate one for the fusing operation. In the experiments, three positions (P_1, P_2, and P_3) were selected by the users on each DVRI wheel, and the system created the corresponding DVRIs by fusing the primary DVRIs accordingly. The resulting DVRIs are placed around the DVRI wheels as shown in Fig. 5(a) and 5(b).

The system was tested for animation generation in the third experiment. In this experiment, a path $P_1 \rightarrow P_2 \rightarrow P_3$ was selected on the DVRI wheel as shown in Fig. 5(c). The system created a series of DVRIs to form a smooth animation along the path automatically. The number of DVRIs contained in the animation is determined by the length of the path and the density of the sampling points. There were 45 new DVRIs created within 5 minutes in this experiment. Several DVRIs from the generated animation along the path $P_1 \rightarrow P_2 \rightarrow P_3$ are shown in Fig. 5(c)).

6.2. Image Quality Evaluation

Finally, we validated the effectiveness of the image quality metric used in the radial graph layout by the fourth experiment on a CT lobster dataset ($256 \times 254 \times 57$). In this experiment, users explored and analyzed the data freely with different rendering parameters in a radial graph (see Fig. 5(d)). When a new DVRI was created, the system inserted it at an appropriate position in a sector according to the rendering viewpoint and its computed image quality value. This insertion operation is fast and usually completed within one second. From Fig. 5(d), we can see that the DVRIs with higher image quality are placed closer to the center of the graph and hence have larger display sizes and so are emphasized.

7. Conclusions

In this paper, we developed a novel palette-style volume exploration interface for creating DVRIs from primary DVRIs in an intuitive way. By imitating the widely-used color wheel, our approach is easy for users to understand. Smooth animation can be easily created by users through selecting a path on the DVRI wheel. To facilitate the detailed exploration of volumetric data, we further proposed a radial graph to arrange the DVRIs automatically based on image quality, so that better DVRIs lie closer to the center and have larger display sizes. Similar to other volume visualization systems that visualize the visualization process [Ma99, JKM01], our system can keep users informed about what is going on by organizing the results in a visual and intuitive way and reduce exploration redundancy. In contrast to existing approaches, the DVRIs on the DVRI wheel are organized based on similarity so that similar DVRIs are placed in closer proximity and users can find their desired results quickly. Moreover, it facilitates opacity TF design on the DVRI wheel by incorporating a high-level TF design method [WQ07].

Our system is flexible and easy to extend. Other high-level TF design methods can also be incorporated into the DVRI wheel interface easily. In future, we plan to integrate a high-level TF design method [SKK06] into our interface, so that the position on the DVRI wheel encodes the visibility of each structure. In addition, the radial graph used in the system for detailed volume exploration can exploit other image evaluation metrics to organize the DVRIs. Formal user study will also be conducted to evaluate the effectiveness of our interface.

(a)

(b)

(c)

(d)
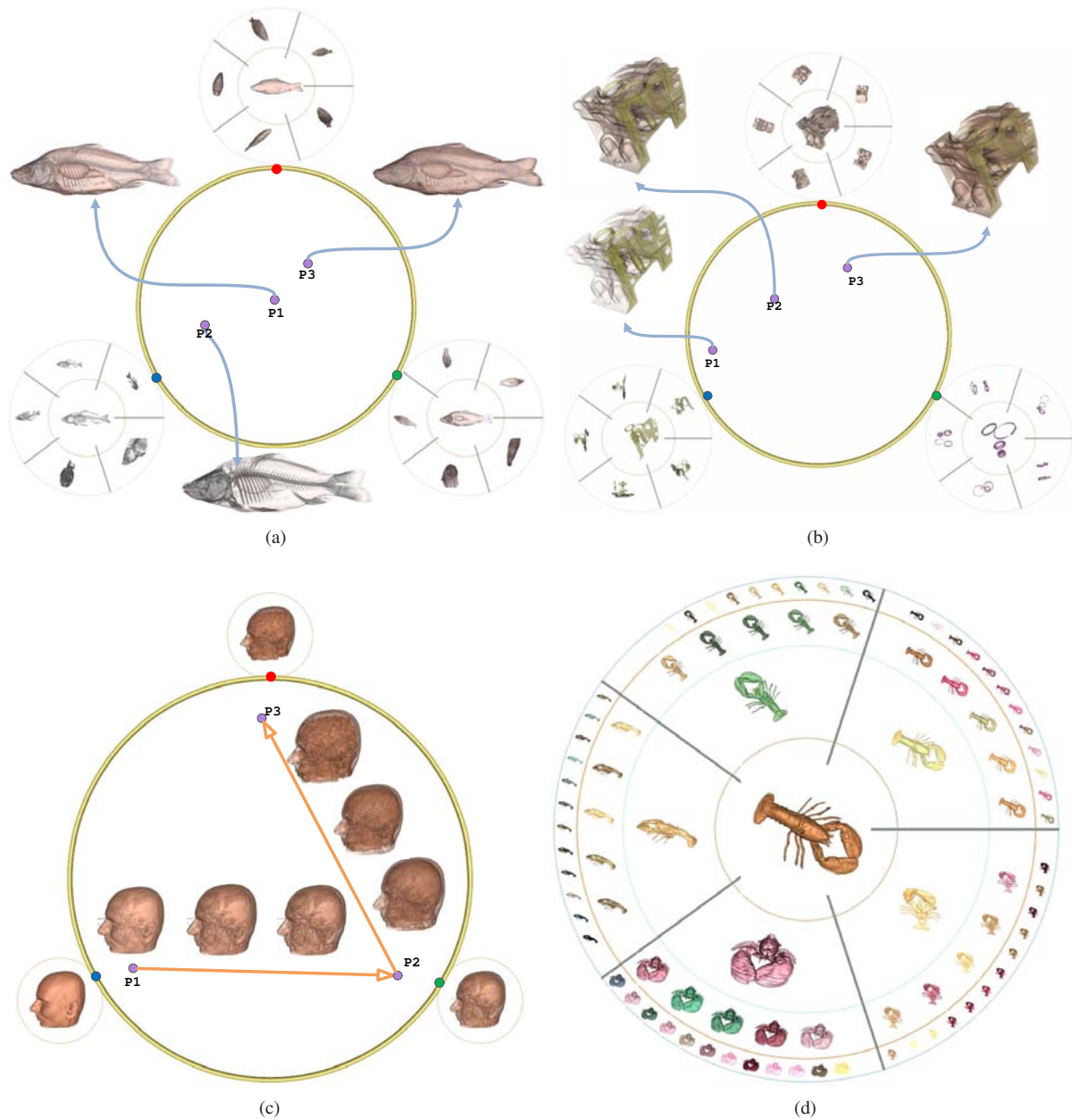
Figure 5: *(a)-(b) Two palette-style DVRI wheels that were captured when they were being used for fusing primary DVRIs. Users selected three points P_1, P_2, and P_3 on each wheel. Based on the distance from the selected points to the primary DVRIs, the system created the corresponding DVRIs by fusing the primary DVRIs. The resulting DVRIs are placed around the DVRI wheels; (c) A palette-style DVRI wheel for creating animation. An animation was generated along the user selected path $P_1 \rightarrow P_2 \rightarrow P_3$; (d) A radial graph for detailed volume exploration. The radial graph was divided into multiple sectors for different viewpoints, and the DVRIs in each sector were sorted according to the image quality such that DVRIs with higher image quality lie closer to the center.*

References

[CXS03] CHENG H. D., XUE M., SHI X. J.: Contrast enhancement based on a novel homogeneity measurement. *Pattern Recognition 36*, 11 (2003), 2687–2697.

[Ead92] EADES P.: Drawing free trees. In *Bulletin of The Institute of Combinatorics and Its Applications 5*. 1992, pp. 10–36.

[JKM01] JANKUN-KELLY T., MA K.-L.: Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Transactions on Visualization and Computer Graphics 7*, 3 (2001), 275–287.

[JKM03] JANKUN-KELLY T., MA K.-L.: Moiregraphs: Radial focus+context visualization and interaction for graphs with visual nodes. In *Proceedings of IEEE Symposium on Information Visualization* (2003), pp. 59–66.

[JKMG07] JANKUN-KELLY T., MA K.-L., GERTZ M.: A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 357–369.

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the IEEE Symposium on Volume visualization* (1998), pp. 79–86.

[KG01] KÖNIG A. H., GRÖLLER E. M.: Mastering transfer function specification by using volumepro technology. In *Proceedings of Spring Conference on Computer Graphics* (2001), pp. 279–286.

[Kin02] KINDLMANN G.: Transfer functions in direct volume rendering: Design, interface, interaction. In *Course notes of ACM SIGGRAPH* (2002).

[Kit] KITWARE INC: The visualization toolkit. www.vtk.org.

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (2002), 270–285.

[KNS04] KREUSELER M., NOCKE T., SCHUMANN H.: A history mechanism for visual data mining. In *Proceedings of Information Visualization* (2004), pp. 49–56.

[LSM06] LUM E. B., SHEARER J., MA K.-L.: Interactive multi-scale exploration for volume classification. *The Visual Computer 22*, 9–11 (2006), 622–630.

[Ma99] MA K.-L.: Image graphs - a novel interface for visual data exploration. In *Proceedings of IEEE Visualization* (1999), pp. 81–88.

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of ACM SIGGRAPH* (1997), pp. 389–400.

[Mit96] MITCHELL M.: *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[NM90] NIELSEN J., MOLICH R.: Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1990), pp. 249–256.

[PLB*01] PFISTER H., LORENSEN B., BAJAJ C., KINDLMANN G., SCHROEDER W., AVILA L. S., MARTIN K., MACHIRAJU R., LEE J.: The transfer function bake-off. *IEEE Computer Graphics and Applications 21*, 3 (2001), 16–22.

[SKK06] SALAMA C. R., KELLER M., KOHLMANN P.: High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1021 – 1028.

[TBET98] TOLLIS I. G., BATTISTA G. D., EADES P., TAMASSIA R.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1998.

[TLM05] TZENG F.-Y., LUM E. B., MA K.-L.: An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 273–284.

[TM05] TORY M., MÖLLER T.: Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics 10*, 1 (2005), 72–84.

[TPM05] TORY M., POTTS S., MÖLLER T.: A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 71–80.

[vBSG06] ŠEREDA P., BARTROL A. V., SERLIE I. W., GERRITSEN F. A.: Visualization of boundaries in volumetric data sets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (2006), 208–218.

[WQ07] WU Y., QU H.: Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 1027–1040.