Collaborative Visualization for Integrated Visual Informatics

Yuriko Takeshima (Senior Assistant Professor), Shigeru Obayashi (Concurrent Professor)

Computer visualization is expected to play a crucial role in studies on transdisciplinary fluid integration. We focus on advanced visual computing approaches for exploring the complex dynamics of various time-varying phenomena, with a special emphasis on promising paradigms for realizing sophisticated fluid informatics. In this section, we introduce the three main themes of our study: a cooperative visualization lifecycle management system, a time-critical rendering of large-scale particle systems, and a topologically accentuated visualization.

2.1. Introduction

Computer visualization is widely used to analyze datasets obtained from numerical simulations and experiments. However, the increasing size and complexity of datasets makes it difficult to obtain appropriate visualization results by trial-and-error. We have been pursuing the development of efficient visualization methods for obtaining effective visualization results. Our research themes are classified into three major groups: a cooperative visualization lifecycle management system, a time-critical rendering of large-scale particle systems, and a topologically accentuated visualization of measurement-integrated simulation datasets.

In Section 2.2, we describ a cooperative visualization environment called TFI-AS/V (Transdisciplinary Fluid Integration-Archive System/Visualization), which not only assists users in selecting the best visualization techniques but also provides them with support for browsing and maintaining detailed provenance information throughout the visualization lifecycle.

In Section 2.3, for stabilizing the frame rates in large-scale particle visualization, we introduc a novel time-critical rendering scheme that combines two methods—a fast method based on shaded texture mapping and a high-quality method using quadratic surfaces. A frame-rate stabilization mechanism based on a PID controller is deployed to change the threshold distance dynamically, in order to regulate the number of high-quality particles.

In Section 2.4, we explain various advanced visualization schemes based on topological structure: topologically-accentuated volume rendering, multi-dimensional transfer function desing, and topologically-enhanced juxtaposition tool.

2.2. Cooperative Visualization Lifecycle Management System

Computer visualization is an indispensable technique, because it enables researchers to realize their objectives through the intuitive analysis of related datasets. However, scientists and engineers who are not necessarily visualization experts may find it difficult to select visualization techniques that best suit their purposes, and hence obtain insightful visualization. Even if they succeed, they need to manage the acquired knowledge separately from the visualization results, analysis process, etc. Consequently, they have to maintain the relevant information by themselves. To address this problem, we proposed a cooperative visualization environment called TFI-AS/V (Transdisciplinary Fluid Integration-Archive System/Visualization) [1]. This system not only assists users in selecting the best visualization techniques but also provides them with support for browsing and maintaining detailed provenance information throughout the visualization lifecycle. The proposed system consists of two subsystems. One is the workflow design support subsystem, which is designed and constructed using an ontological framework in order to enable users to design their visualization workflows. The other is the provenance management subsystem, which is intended to provide a scalable mechanism that enables multiple users with a common visual analysis objective to record, trace, and reuse their visual exploration workflows along with the acquired knowledge and results.

In this section, we provide a detailed explanation about our workflow design support subsystem, which is called GADGET/FV (Goal-oriented Application Design Guidance for modular visualization environmenTs/Flow Visualization).

2.2.A. Introduction

With the advent of supercomputers, large-scale and complex (time-varying and multi-dimensional/multivariate) datasets have been readily generated, and thus visualization has played a crucial role in effective analysis of these datasets. For rapid prototyping in many disciplines, mainly due to their archivability and extensibility, modular visualization environments (MVEs)[2] are used primarily. However, every MVE usually forces the users to design their own workflows by themselves. This requires their experiences and expertise on how to utilize hundreds of built-in modules by taking into account their usability and interdependecy for constructing optimal visualization workflows. Therefore, it is still difficult for scientists and engineers, who are not always visualization experts, to make full use of the MVEs.

To address this problem, we have developed a sophisticated MVE series, which is called GADGET (Goal-oriented Application Design Guidance for modular visualization EnvironmenTs)[3–5]. In this section, we focus on a novel variant, called GADGET/FV (Flow Visualization)[5], for the design of visualization workflows in fluid science, where sufficient support for higher-order (i.e., vector and tensor) fields as well as scalar fields is considered. The GADGET/FV system interactively helps the users design appropriate workflows on AVS/Express, according to a visualization goal to be specifiable with an extended version of the Wehrend Matrix[6]; such properties as dimension and mesh type of a given target dataset; and temporal efficiency versus accuracy requirements. Furthermore, the system allows the users to make inquiries about analogous workflow designs maintained in the case repository. This special feature to facilitate visualization workflow design is herein referred to as "design by example." In [7], the pursuit of ontological organizations and opening of case repository to the public are regarded as the most important subjects that should be taken into consideration in the future visualization systems.

2.2.B. Ontological Framework for Flow Visualization Design

The Wehrend Matrix[6] classifies existing visualization techniques by pairing words from two types of vocabulary lists, i.e., action and target. The classification of actions distinguishes problems in terms of the representation visualization goals, whereas the classification of targets groups techniques based on the nature of objects in the target domain. Keller and Keller[8] actually classified more than one hundred visualization techniques using the following sets of action and target:

action = { "identify", "locate", "distinguish", "categorize",
"cluster", "rank", "compare", "associate", "correlate" }

target = { "scalar", "nominal", "direction", "shape", "position",
"Spatially Extended Region or Object", "structure" }.

Note that the two actions "associate" and "correlate" require more than one target, and that any pair of action and target does not necessarily have corresponding visualization techniques. For example, "Colored arrow plots" is a classical technique to"correlate scalar and direction".

Our flow visualization ontology is defined by elaborating the Wehrend Matrix for fluid science.

(a) Data types
A data type relates deeply to the selection of visualization techniques. It is necessary to consider such properties as dimension and mesh type of a target dataset, both of which are key factors to determine the applicability of visualization techniques. Generally, these types of information can be extracted automatically from the header of a target dataset.

(b) Flow properties
Flow fields can be classified into two types: steady and unsteady. We analyze only a single-time dataset for steady flow since the flow field is not time-varying. On the other hand, for unsteady flow, we can choose either a snapshot dataset or a time-series dataset. Therefore, our ontology takes account of flow property, whose value is steady flow, unsteady flow (snapshot), or unsteady flow (time-series). Visualization techniques to be used clearly differ according to these flow properties.

(c) Display styles
Display style settings, snapshot images or animations, should also be specified in our flow visualization ontology. Note that even if a given flow is steady, animations can be created by changing some visualization parameter values.

(d) Visualization goals
The vocabularies of action and target of the Wehrend Matrix are refined by reexamining flow visualization examples included in Keller and Keller's Book[8] as well as by reflecting general surveys on flow field analysis:

action = { "associate", "compare", "correlate", "identify", "locate", "superimpose" }

target = { "scalar", "vector" },

where the three actions "associate", "correlate", and "superimpose" require more than one target. Although the above target looks too simplified compared with the target of the original Wehrend matrix, finer visualization targets will be able to be specified by introducing derived fields, as shown in the next item.

Table 1 Derived fields specifiable with GADGET/FV.

target	derived field
scalar	original data
	magnitude
	magnitude of vorticity vector
	magnitude of gradient vector
vector	original data
	vorticity vector
	gradient vector

Table 2 Examples of design directives and corresponding visualization techniques.

Attributes	**Example 1**	**Example 2**
dimension	3	3
meshtype	rectilinear	rectilinear
flow property	steady	unsteady (time-series)
output style	snapshot	animation
action	identify	compare
target	vector	scalar
derived field	vorticity vector	original data
technique	arrow plots streamlines	cross-sections multiple isosurfaces

(e) Derived fields

In flow field analysis, it is common to derive and visualize such fields as gradient and vorticity vectors from the original velocity vectors. Therefore, our ontology lets the sub-target fields to include the derived fields listed in Table 1, where original data means the original field of a given dataset. As long as the users specify the sub-target field that they want to visualize, they will not have to compute them in advance, which becomes easier to visualize various derived fields from the given dataset.

(f) Adaptation to large-scale datasets

When considering large-scale data visualization, our ontology should consider various aspects of computing, including domain selection; downsizing; and parallelization. In particular, Flynn's taxonomy could be a good candidate to select the parallel visualization architecture to be used.

The current GADGET/FV system returns the list of appropriate visualization techniques according to the design directives with the above-mentioned five items from (a) to (e). Two examples are listed in Table 2. Note that the same visualization techniques can be selected although their design directives are different. Generally, design directives and visualization techniques have many-to-many relationships.

2.2.C. Visualization Workflow Design Support

This section describes how the GADGET/FV system assists the users in designing their visualization workflows based on the ontological framework delineated in Section 2.2.B. Figure 2.1 illustrates the user processing and data/control flows in the system. Hereafter, each of the design stages is explained in detail along the main processing flow of the system.

Stage 0: Setteing user's skill level

The GADGET/FV system changes its protocol to assist the users adaptively according to the skill level they declare. Indeed, at Stage 3, shown to beginners and intermediate users are only the windows for visualization parameter setting and visualization results. Some hints are additionally given for beginners to adjust visualization parameter values by themselves. On the other hand, advanced users can take a look at the actual definition of visualization workflow for the selected visualization technique, and are provided a privilege to customize it freely on the MVE at the backend.

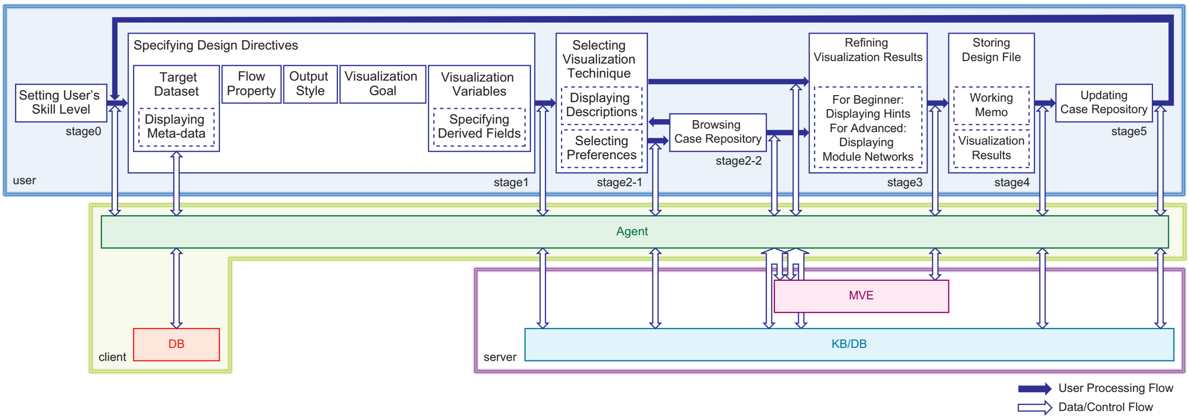


Figure 2.1 Main processing flow in GADGET/FV.

Stage 1: Specifying design directives

The design directive is defined with the extended Wehrend Matrix:

(1) The user specifies a dataset to visualize. The system then loads the specified dataset from the database on the client, and extracts necessary header information, such as dimension, mesh type, and field labels of the dataset. Once it was extracted, the user is allowed to check the meta-information in a separated window.

(2) The user sets the flow property.

(3) The user chooses the output style.

(4) The user specifies the visualization goal with the combination of action and target.

(5) The user selects specific fields to be used as target. The system displays the field labels which have already been acquired in (1). If the target is scalar, the user chooses a single variable. If it is vector, the user chooses all the necessary component variables. At this moment, the derived fields can also be selected.

Stage 2-1: Selecting visualization technique

The system returns the ordered list of appropriate visualization techniques which are retrieved from the knowledge-base by using the design directive as a key. At this point, the user chooses one preferable technique from the technique list. The descriptions of the chosen visualization technique can also be shown in a separated window. Moreover, the user is allowed to set his preference on temporal efficiency or accuracy of visualization.

Stage 2-2: Browsing case repository

The case repository is opened to the users. The system retrieves the successful visualization examples of the selected technique from the system KB/DB on the server, and shows the resulting images / animations together with associated information such as created date and time, used dataset and user name. If needed, the user can reuse the underlying workflow of the selected case for his/her own visualization design.

Stage 3: Refining the visualization results

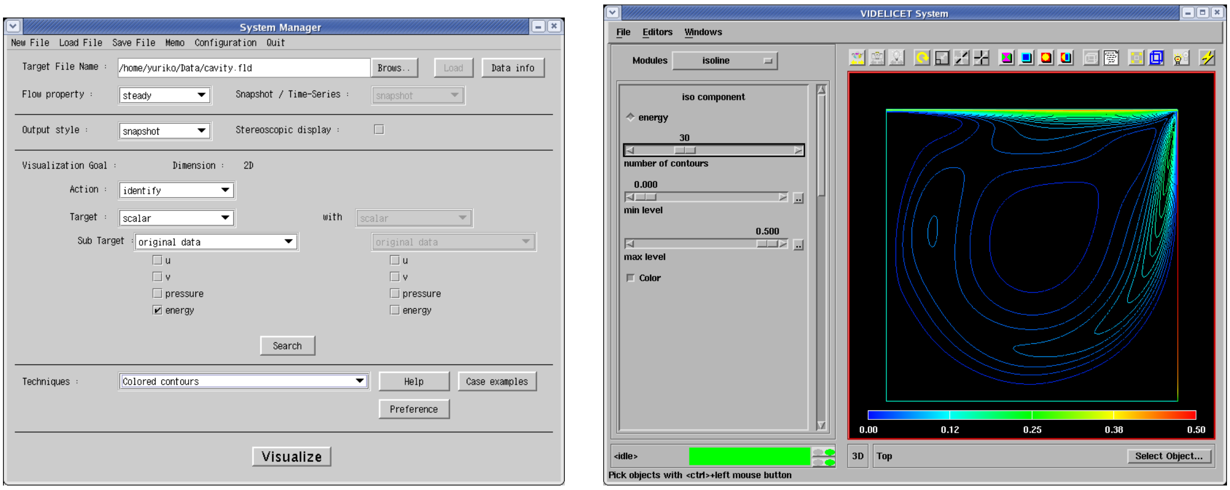
The system forwards the designed workflow and the given dataset to the MVE, and returns the visualization result to the user. The user is allowed to improve the result by changing the visualization parameter values interactively.

Stage 4: Storing design file

The user can store all the information related to the visualization design in the form of visualization workflow design file, which actually contains filename of the dataset, design directive, up-to-date visualization parameter values and visualization results. The user can reuse, or modify if needed, the designed workflow to generate relevant visualization results, for example, for the purpose of systematic parameter study. The visualization workflow design file also provides the functionality to associate free-format working memos with the corresponding visualization resources. Such consolidation is expected to contribute the realization of an effective e-science environment.

Stage 5: Updating case repository

If the created visualization workflow design is permitted to be opened to others, it can be registered by the user into the case repository. Prolonged use of the GADGET/FV system by many users makes the case repository more diversified and versatile.



(a) (b)

Figure 2.2 Visualization of cavity flow dataset using GADGET/FV system: (a) System interface and (b) Visualization result.

2.2.D. Applications

This section illustrates the usefulness of the GADGET/FV system by applying it to the design of visualization workflows for practical flow field datasets. The current prototype of the GADGET/FV system has been implemented on a HP workstation (CPU: Intel Xeon 3.60GHz; Memory: 4GB) as the server, AVS/Express7.0 as the MVE, and Oracle Database 10g as the knowledge-base management system.

We show herein two examples of visualization workflow design: One follows the fundamental procedure, and the other uses the case repository to demonstrate the concept of "design by example".

Case 1: Fundamental procedure

Suppose that a user attempted to visualize a 2D cavity flow dataset (128×128). Figure 2.2 shows a screenshot of the system interface and the visualization result. He declared his skill level by pushing the Configuration button in the menu bar, and beginner level was chosen in this case. Next, he specified the dataset to visualize, and loaded it to the system by pushing the Load button. If the Datainfo button was pushed at this moment, the meta information of the loaded dataset was displayed. As the loaded dataset represents a steady flow, he set steady for Flow property. Since the dataset contains a single-time data, the selection of Snapshot/Time-series became inapplicable and greyed out. Then, he set snapshot for Output style. In order to investigate the distribution of energy of the given dataset, he specified the visualization goal so as to identify scalar, where original dataset was set for Sub-target, and the energy label was checked.

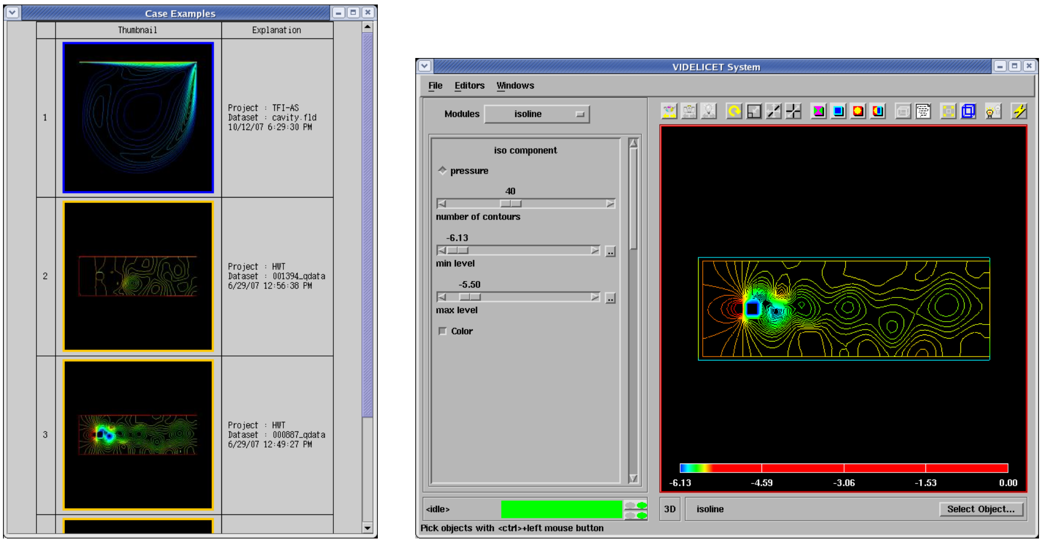
Next, pushing the Search button invoked the retrieval of corresponding techniques from the knowledge-base, and made the list displayed. In this case, 19 visualization techniques were found, including "pseudo-color coding", "colored contours", "colored height field" and "colored dots". After selecting "colored contours" and pushing the Visualize button, the hint window was popped up to show how to set control parameters of the colored contours module, including the number of contours and the field interval. The user was allowed to interactively change the value for each of the visualization parameters to obtain better results.

Finally, the user was allowed to parameterize the entire process into a visualization workflow design file. Also, he permitted the file to be registered in the case repository.

Case 2: Design by Example

Suppose that another user attempted to use the GADGET system for visualizing the 2D pressure field, which was obtained from the measurement-integrated simulation in the Hybrid Wind Tunnel project [9]. The grid size of the target dataset here is 60×23. Since intermediate was chosen as the skill level this time, the hints window was no longer displayed.

Since the visualization task can be characterized with unsteady for Flow property, snapshot for Snapshot/Time-series, and the same output style and visualization goal (except for the scalar field label) as in Case1, the identical list of recommended visualization techniques was displayed. After he chose the same visualization technique colored contours from the list, and then pushed the Case example button this time, he was allowed to browse successful examples where the same technique was applied to various datasets in the case repository window, as shown in Figure2.3(a). Note that the result obtained in Case 1 can be seen in the case repository window.



(a) (b)

Figure 2.3 "Design by Example": (a) Case repository window and (b) Visualization result.

At this moment, the user was allowed to choose either of two processing procedures. One is to finish browsing the case repository, and then continue the process just like in Case 1. The other is to right-click on a preferable thumbnail to invoke the corresponding workflow for the target dataset by inheriting the conditions of the selected example as much as possible. The latter way is referred to as "design by example". Figure 2.3(b) shows the final result, which was obtained by the "design by example" with the case of Case 1.

2.2.E. Conclusion and Future Work

In this section, we proposed a contemporary MVE, which builds on an ontological framework to support the design of flow visualization workflows. An extended Wehrend Matrix with derived flow fields is a key integral part of the flow visualization ontology. Browsing the case repository led to an effective shortcut to design their own workflows, which is called "design by example". The feasibility of the system was empirically proven through an experimental use to visualize a measurement-integrated simulation dataset.

At present, we plan to utilize the GADGET/FV system as one of the primary components to develop the visualization lifecycle management system [10], where the GADGET/FV system is expected to allow the users to start with proper initial designs, which facilitates the subsequent versioning of the workflows.

In our future R&Ds, we need to refine the Wehrend Matrix vocabularies to specify the visualization goals in fluid science in a more enriched manner. For example, the ontological framework could be extended so as to support the design of visualization workflows for large-scale and multidimensional / multivariate datasets. If the space partition information was available, the system could take full advantage of the information to decide the actual strategy to perform domain-parallel visualization. If the system accepted multiple goal specification, multifield visualization could be reduced into concurrent execution of single field visualizations. Furthermore, the system could provide the user with some suggestions on the problem parallelizability by taking account of constraints on computing resources together with the problem complexities.

We need to expand the range of user consultation to visualization parameter settings, because choosing proper values for control parameters is a key to the success of any visualization. To this end, it would be of particular interest to incorporate to the current system, our prior work on topologically-based visual data mining [11]. Also, it would be very challenging to extend the flow visualization ontology so as to deal with multimodal analysis (e.g., sonification and haptizetion) for multivariate datasets.

2.3. Time-Critical Rendering of Particle Systems

in visual analysis, it is important to maintain interactive and stable rendering frame rates, because it is known that unstable frame rates may cause stress and distraction to the users. For stabilizing the frame rates in large-scale particle visualization, we proposed a novel time-critical rendering scheme [12] that combines two methods—a fast method based on shaded texture mapping and a high-quality method using quadratic surfaces. A method to render each particle is decided on the fly according to the viewing distance; the high-quality method is chosen only when the distance is smaller than a threshold, to allow the user to observe the region of interest closely. A frame-rate stabilization mechanism based on a PID controller is deployed to change the threshold distance dynamically, in order to regulate the number of high-quality particles.

In this section, we introduce our time-critical rendering scheme for particle visualization.

2.3.A. Introduction

Interactive visualization of large-scale datasets of particles on a standard PC is difficult because the rendering time increases as more particles need to be rendered. To alleviate this problem, various methods have been proposed, including a high-speed rendering method [13] and a selective method that renders only a region of interest (ROI) in a given dataset [14]. However, it is known that a trade-off between speed and quality exists in these rendering methods, and it is almost impossible for a single method to render high-quality particles at a sufficiently high speed even when a modern GPU environment is used. In addition, the rendering speed should be stabilized to allow the users to perform a stress-free interactive analysis. To this end, we propose a novel time-critical rendering scheme that renders as many accurate particles as possible while maintaining an interactive frame rate. To control the rendering speed and quality, we deploy two rendering methods—a fast rendering method based on shaded texture mapping and a high-quality method using quadratic surfaces [15], in a combined manner. Under an assumption that a ROI exists near the viewpoint, we select an actual rendering method for each particle on the basis of the viewing distance. If the viewing distance of a target particle is smaller than a threshold, the particle is rendered using the high-quality method; else it is rendered using the fast method. In general, the users frequently change their viewpoint to peer into the data domain. The fixed threshold distance may give rise to unstable frame rates, because the total rendering speed depends on the number of particles to be rendered by each of the methods. For stabilizing the frame rate, a mechanism based on proportional-integral-derivative (PID) feedback controller is introduced for our time-critical rendering scheme. We confirm its effectiveness through experiments using molecular dynamics simulation datasets.

2.3.B. Rendering Method

To determine relative position among particles accurately, the rendering method should be able to represent the space-filling model. However, the traditional methods used for geometrically rendering many spherical particles may suffer from spatiotemporal computation complexities. To alleviate this problem, a GPU-based rendering method is used for quadratic surfaces [15], where the depth value and color of each pixel are evaluated accurately from the intersection point between a ray cast from the viewpoint and a spherical surface that represents a target particle. As a result, it can provide the same rendering quality as the traditional geometrical rendering method. However, even with the quadratic surface method, it becomes hard to maintain the interactivity as the number of particles increases. Therefore, we build on the model of human visual acuity to let our scheme employ a fast rendering method as well, which is based on point-based shaded texture mapping for particles outside of the ROI. The shaded texture is created as a pseudo-texture by alpha blending a portion of the template texture that represents shade and highlight with a base spherical texture. Although the accuracy of this method is not as high as that of the space-filling model, it can render a particle faster than the quadratic surface method because it eliminates the need for preparing a different shaded texture for each particle depending on the light source and viewpoint.

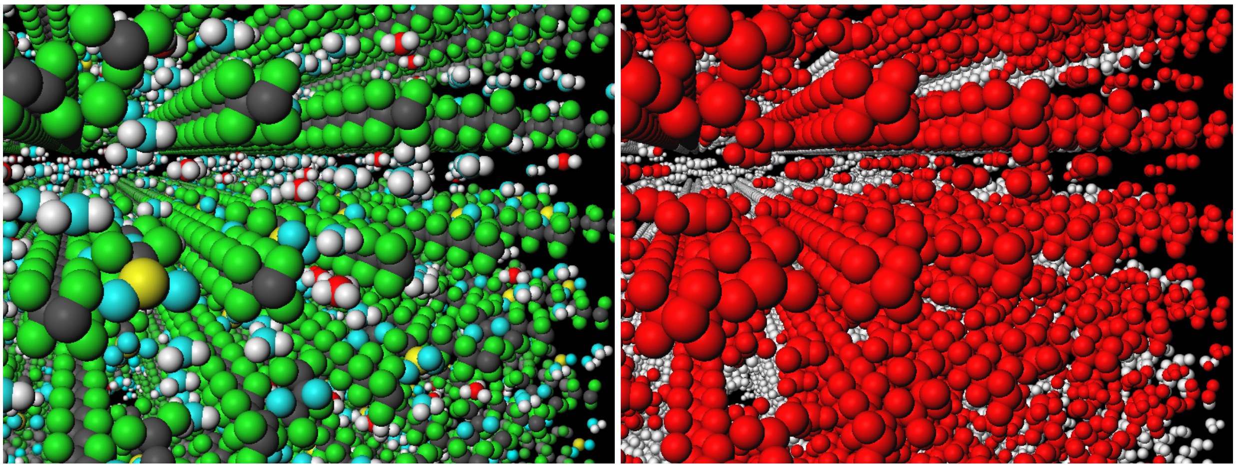
2.3.C. Stabilizing Frame Rates Using PID Controller

The method to render each particle is decided on the fly according to the distance from the viewpoint. We propose a frame-rate stabilization mechanism using a PID controller. The rendering timing is stabilized by controlling the number of high-quality particles on the basis of the error which is the gap between actual and expected values in the rendering timing. The proportional, integral, and derivative values are defined as the response to the current error, the response based on the sum of recent errors, and the response based on the rate at which the error varies, respectively. If the error at timestep t is defined as $\Delta y(t)$, the variation in the number of particles rendered by the high-quality method, $\Delta x(t)$, can be represented as

$$\Delta x(t) = K_p \Delta y(t) + K_i \int_0^t \Delta y(\tau) d\tau + K_d \frac{d\Delta y(t)}{dt}, \tag{2.1}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. If the number of particles rendered by the high-quality method is controlled strictly on the basis of the viewing distance, the computational cost increases because the particles need to be sorted by their distances at each timestep. In this study, the switching the rendering methods is performed as follows. The two extreme rendering timings when all the particles are rendered by either the fast method or the high-quality method are pre-computed. These rendering timings are denoted as f_{tm} and f_{qs}, respectively. The threshold distance at time step t, $dist(t)$, is determined according to the following formula:

$$dist(t) = dist(t-1) - (dist_{max}(t) - dist_{min}(t)) \times \Delta x(t), \tag{2.2}$$



(a) (b)

Figure 2.4 Visualization result of the molecular dynamics simulation dataset for bubble nucleation; (a) rendering result and (b) color-coded particles rendered by the quadratic surface method (red) and the shaded texture mapping method (white).

Table 3 Statistical analysis of frame rates.

scheme	default scheme		our scheme	
no. of particles	27,480 (data A)	89,100 (data B)	27,480 (data A)	89,100 (data B)
average frame rate (fps)	26.25	8.11	31.34	10.67
standard deviation	12.68	3.93	1.31	0.30

where $dist_{min}(t)$ and $dist_{max}(t)$ denote the minimum and maximum distances between the viewpoint and the bounding box of the target dataset, respectively. If the distance of a target particle from the viewpoint is smaller than the threshold, the particle is rendered by using the high-quality method; else it is rendered by using the fast method. K_p, K_i, and K_d were set empirically to $0.95 \times (f_{tm} - f_{qs})$, $0.50 \times (f_{tm} - f_{qs})$, and $0.25 \times (f_{tm} - f_{qs})$, respectively.

2.3.D. Experiments

We evaluated the effectiveness of our time-critical rendering scheme through experiments with molecular dynamics simulation datasets for bubble nucleation [16]. Our system was implemented in C++, OpenGL, and GLSL (OpenGL Shading Language) on a standard PC (CPU: Intel Xeon 3.80GHz, RAM: 2GB, GPU: NVIDIA Quadro FX4500).

Fig. 2.4 shows a visualization result of the dataset and a color-coded image of the particles rendered by the two methods. From Fig. 2.4, we can clearly identify the spatial relationship among the particles since those included in the ROI were represented by large spheres in the image, and thus rendered accurately. In addition, the rendering accuracy of the fast method for the distant particles had no problem, as those were delineated by small enough spheres.

Next, we examined the variation in the rendering speed for two datasets with the following numbers of particles: 27,480 (data A) and 89,100 (data B). In this experiment, we analyzed a walkthrough animation with the following three scenarios: (1) from frame 0 to frame 250, the user searched for the ROI by rotating the dataset; (2) from frame 251 to frame 350, the user moved his viewpoint toward the ROI; and (3) from frame 351 to frame 500, he observed the ROI in more detail by moving the viewpoint slightly. Fig. 2.5 shows a comparison between the frame rates obtained by applying a default scheme that uses a fixed threshold distance and those by applying our scheme to the walkthrough animation. When all the particles were rendered by the shaded texture mapping, the frame rates were 49.28 fps (data A) and 16.67 fps (data B), and when they were rendered by the quadratic surface method, the frame rates were 4.33 fps (data A) and 1.65 fps (data B). Furthermore, the target frame rates for the PID controller were 31.30 fps (data A) and 10.66 fps (data B), which were set to $0.6 \times (f_{tm} - f_{qs}) + f_{qs}$. In addition, the threshold distance for the default scheme was fixed as half the length of the diagonal of the bounding box. From Fig. 2.5, we can recognize that the frame rates of the default scheme changed drastically for the two datasets. On the other hand, the frame rates of our scheme vibrated initially whereas they finally converged to the target frame rate.

Table 3 tabulates the average values and the standard deviations of the frame rates after frame 20 when the PID controller took effect. The expectation values of the default scheme and our scheme were the average frame rate and the target frame rate, respectively. The frame rates of our scheme were stable at the target frame rate. Furthermore, the standard deviations for data A were higher than those of data B, because the difference of the frame rates for data A was larger than that of data B even if the difference of rendering timings were analogous.

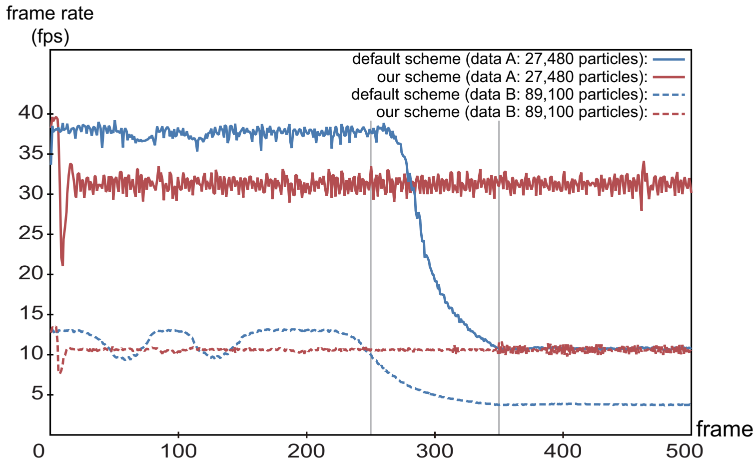


Figure 2.5 Comparison between frame rates obtained by applying a default scheme and our scheme for a walkthrough animation.

2.3.E. Conclusion

In this research, we proposed a time-critical rendering scheme for large-scale particle visualization; our scheme uses a combination of two rendering methods. For stabilizing the frame rate, a new stabilization mechanism based on a PID controller is introduced. The feasibility of our scheme was empirically proven by performing experiments using molecular dynamics simulation datasets for bubble nucleation.

The gains of the PID controller used in this study were prefixed. However, these should be set using heuristic tuning methods such as Ziegler-Nichols method and Cohen-Coon method. Although the influences of computing environment on the rendering speed were determined in terms of measured rendering timings, these can be estimated more precisely on the basis of the specifications of the environment and running tasks.

2.4. Topologically-Accentuated Visualization

The visualization parameter values are critical because they have a significant influence on the knowledge acquired from the visualization results. The most common approach is to determine these values by trial-and-error. However, such an approach does not guarantee appropriate visualization results. To address this problem, we proposed a topology-based scheme for setting the visualization parameter values. In our research, topological structures are extracted by tracing topological changes in evolving isosurfaces according to scalar field values, and then, a level-set graph called a volume skeleton tree is constructed. We have developed various advanced visualization schemes based on topological structure. After describing our differential topology analysis, we introduce our method of a topologically-accentuated volume rendering, multi-dimensional transfer function desing, and topologically-enhanced juxtaposition tool.

2.4.A. Differential Topology Analysis

Prior to calculating topological attributes, we extract a level-set graph of a given volume dataset using the topological volume skeletonization algorithm [17]. The level-set graph, called volume skeleton tree (VST) [18], allows us to evaluate the topological attributes of each voxel by illuminating both the global and local features of the volume dataset.

A node of the VST represents critical point that has the change either in the number of connected isosurface components or in the genus of each of the isosurface components. Critical points are classified into four groups: maxima (C_3), saddles (C_2, C_1), and minima (C_0), which represent isosurface appearance, merging, splitting, and disappearance, respectively, as the scalar field value reduces. Here, an index of a critical point represents the number of negative eigenvalues of the Hessian matrix there. A link of the VST represents an topology-preserving transition of an isosurface connected component. A link is defined solid if its isosurface component expands as the scalar field value reduces while hollow if it shrinks.

The isosurface merging at C_2 and splitting at C_1 have both four topological transition paths with different isosurface spatial configurations, as shown in Fig. 2.6. In what follows, the VST uses the notation for the critical points with its own connectivity as illustrated in Fig. 2.6, where the solid incident link represents a solid isosurface while the broken link represents hollow. Saddle points of $C_i (i = 1, 2)$ are classified into 3-C_i and 2-C_i, according to their degree (valence). For later convenience, all the boundary voxels are assumed to be connected to the virtual minimum having $-\infty$ as its scalar field value [18]. Note that the link incident to the C_0 node is solid when the node is the virtual minimum as shown in Fig. 2.6. An example of the VST together with isosurface transitions is illustrated in Fig. 2.7.

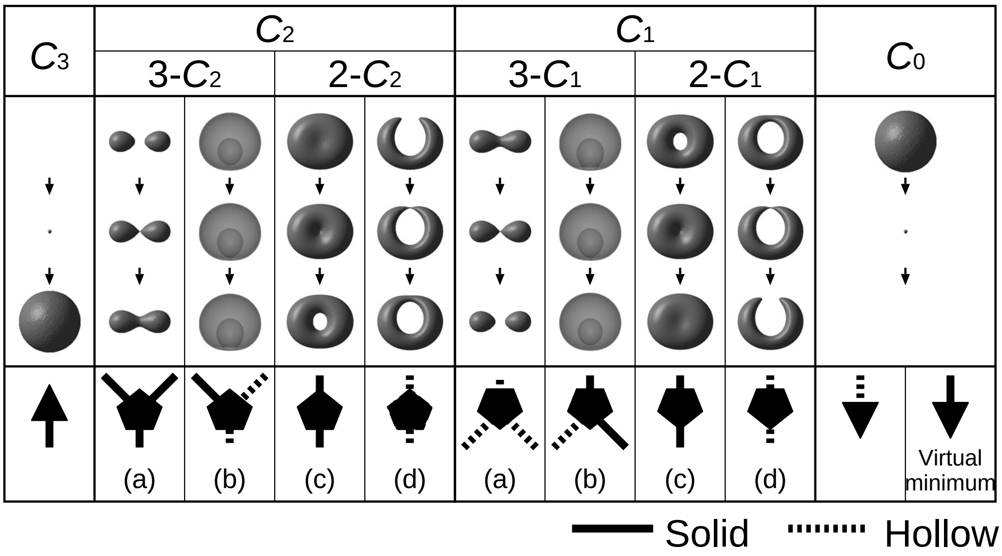


Figure 2.6 The connectivity of a critical point of each type in the VST.

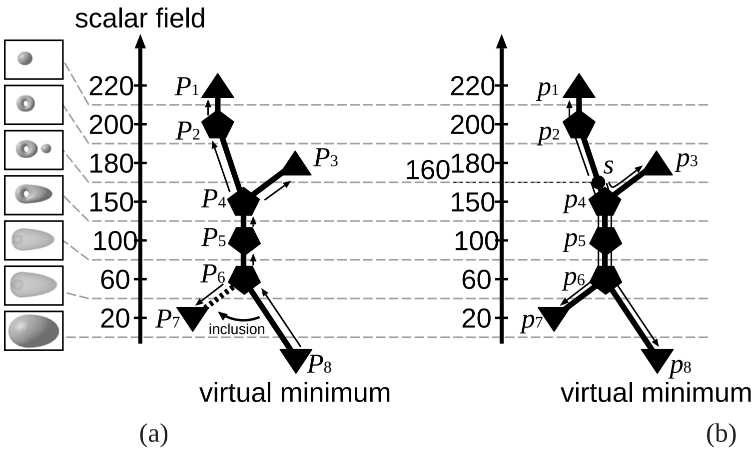


Figure 2.7 The process of calculating (a) the inclusion levels and (b) isosurface-trajectory distances.

In our implementation, a node has its coordinates and scalar field value, and a link has its genus and index of adjacent nodes.

The VST has much in common with the contour tree [19] in the mathematical sense. However, the VST decomposes a multiple (degenerate) critical node into simple ones to extract the global structures such as nested structure of isosurfaces, whereas the contour tree keeps critical points of multiple degrees directly without resolving them into simple ones.

2.4.B. Topologically-Accentuated Volume Rendering

The rapid development of computer hardware and Direct volume rendering, one of the major volume visualization methods, has been further popularized with the recent advent of commodity programmable GPUs. Direct volume rendering can delineate key features of a volume if relevant parameters are judiciously chosen. These parameters include transfer functions, viewpoint location, and illumination. Transfer functions are used to convert physical field values to color and opacity. As this process is independent of how to project a given volume, transfer functions can be specified in preference to viewpoint and light sources. Locating a good viewpoint is as important as specifying good transfer functions because the choice of viewing direction drastically varies the amount of information delineated in the resultant images, by reflecting to what extent the volume features, which are classified with the prespecified transfer functions, are occluded in the direction of projection. Locating optimal light sources also plays a crucial role for shaded volume rendering since it significantly influences the spatial perception of the volume features observed from the prespecified viewpoint. We have ever consistently built upon differential topology to come up with several methods for optimizing volume rendering parameters. It is one of the common benefits of the topological analysis that it can allow the observers to comprehend the local and global features of a given volume simultaneously.

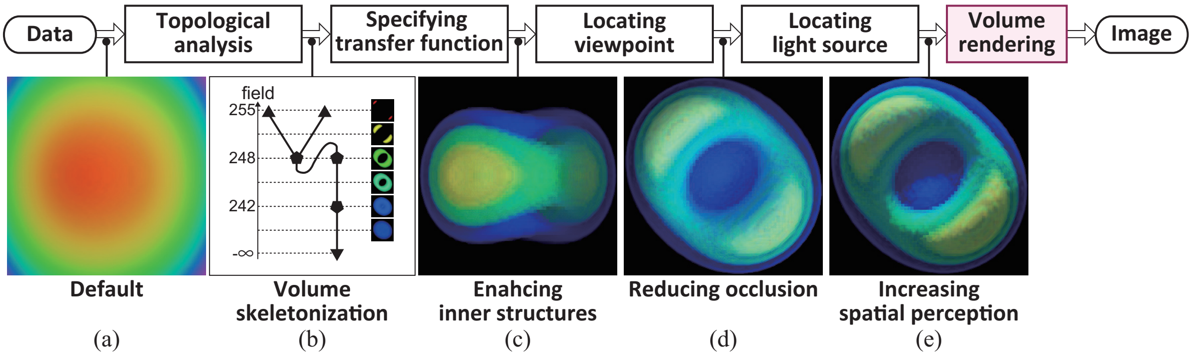


Figure 2.8 The outline of topologically-accentuated volume rendering

The outline of our topologically-accentuated volume rendering which includes incremental visual effects obtained from an analytical volume is shown in Fig. 2.8. Fig. 2.8(a) shows a visualization result with a traditional linear transfer function. In reference to the VST (Fig. 2.8(b)), an opacity transfer function can be designed so as to accentuate representative isosurfaces, each of which is extracted midway in the field interval of a VST link. By comparing the images of Fig. 2.8(a) and (c), it is observed that the inner structure of the volume is uncovered effectively with the topologically-accentuated opacity transfer function [18] An optimal viewpoint is obtained as the position from where the extracted topological features are projected with minimal occlusion onto a screen [20]. To this end, we seek a position so that the weighted sum of the view entropy values of the representative isosurfaces is minimized. A plausible set of weights can be specified by the above opacity transfer function. As shown in Fig. 2.8(d), the resultant image projected from the optimal viewpoint reveals the inner structure of the volume in a more comprehensible manner than the one in Fig. 2.8(c). Fig. 2.8(e) shows the well-contrasted volume shaded image of the analytical volume using a single, optimally-located, parallel light source [21].

2.4.C. Designing Multi-Dimensional Transfer Functions with Topological Attributes

The advent of multi-dimensional transfer functions is one of the latest major achievements in the volume visualization research. As opposed to the traditional one-dimensional transfer functions that only consider a voxel's scalar field value, the multi-dimensional transfer functions assign auxiliary attributes to the voxels to construct their sophisticated parametric domains. For example, if we take into account additional attributes, such as higher-order gradient fields, as well as the original scalar field, we can generate more comprehensible visualization images, where the relative geometric positions and subtle differences among the volumetric features are revealed in an accentuated manner. Such multi-dimensional transfer functions have played an important role in gaining clear insights into a target volume data, especially for the cases without any prior knowledge about the data.

In actual situations, however, observers usually might want to exploit their background knowledge about a target volume data as the clues to perform detailed analysis of the data. For example, when visualizing volumes obtained by scientific simulations, they can utilize their own knowledge about the simulation settings to extract the global characteristics of the volumes and to locate regions of particular interest. If they are allowed to design multi-dimensional transfer functions using proper attributes so as to encapsulate such advance knowledge, they can readily yield visualization results to fulfill their purposes.

Nevertheless, nearly all attributes for the conventional multi-dimensional transfer functions are based on local features, such as differentials and curvatures, and cannot capture the global structure of the volume contrary to the observer's purposes. As such, the observers are forced to design their transfer functions by juxtaposing the explicitly-described local features together with the global structure in their mind, and thus leading to cognitive load as well as qualitative limitations of resultant visualization images.

We therefore introduced a new set of topological attributes to establish a new framework that is intended to realize objective-based assistance, especially for visualizing simulated volume datasets [22]. In our framework, transfer functions are designed so that feature isosurfaces which represent topological features of a given volume dataset are emphasized. However, only the feature isosurfaces selected according to observer's purposes instead of all the feature isosurfaces are emphasized using topological attributes. Topological attributes proposed herein are derived from the level-set graph of a given volume, which delineates the topological evolution of an isosurface with respect to the scalar field. The level-set graph can capture not only the local features but also the global structure of the volume because each node locates the local topological change in the evolving isosurface, and its link the global connection in between. To the best of the authors' knowledge, topological attributes derivable from the level-set graph can be viewed as the only attributes that can reflect the global structure as well as local features of the volume, and can be used to encapsulate the observer's intentions fully into the multi-dimensional transfer functions. Needless to say, these topological attributes are expected to provide more valuable analysis clues than the conventional local feature-based

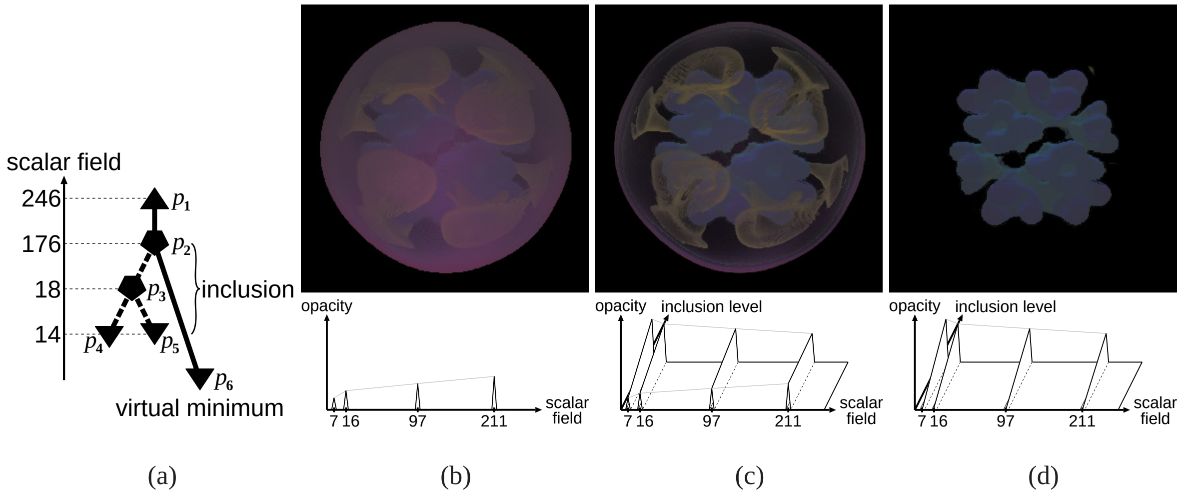


Figure 2.9 Visualizing simulated implosion in laser fusion: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity transfer function, (c) with 2D opacity transfer function depending also on the inclusion level, and (d) with 2D opacity transfer function that visually extracts inner structures.

attributes when visually exploring unknown volume datasets along with background knowledge.

Topological attributes based on the VST can provide a systematic means of emphasizing the underlying volume features, including nested structures of isosurfaces, configuration of the isosurface trajectories, and transitions of isosurface's topological type. We develops several combinations of these topological attributes together with the associated transfer function designs.

Case 1: Revealing isosurface nested structures

Some volume datasets may be characterized with their isosurface nested structures. Fig. 2.9 visualizes a snapshot volume for 3D fuel density distribution excerpted from a time-varying dataset simulating the process of implosion in laser fusion [23], where small bubble-spike structures evolve around a contact surface between a fuel ball (inner) and pusher (outer) during the stagnation phase. The fuel-pusher contact surface can be identified with an isosurface extracted by observing the rapid gradients of the fuel density field. We can learn from the specific simulation setting that the extracted isosurface has two nested connected components, and the contact surface of our interest is occluded by the other outer component residing in the pusher domain, which is nothing but a phantom surface induced by the action-reaction effect [23].

Fig. 2.9(a) shows the VST for the implosion dataset, where the skeletal structure of the complex fuel density distribution has been extracted with an intentional control of VST simplification. A glance at the VST around the field interval $[14, 176]$ finds a nested structure where connected isosurface components corresponding to the links $\overline{P_2P_3}$, $\overline{P_3P_4}$, and $\overline{P_3P_5}$ are included by another connected isosurface component corresponding to the link $\overline{P_2P_6}$.

A volume-rendered image is shown with the topologically-accentuated 1D opacity transfer function in Fig. 2.9(b), from which we can see that after the scalar field itself has been topologically-accentuated, we still suffer from a problem that the inner isosurface components of interest for the observer are indeed occluded by the outer spherical isosurface component. Contrary to that, as shown in Fig. 2.9(c), if we devise the 2D opacity transfer function which depends on the inclusion level as well to assign a lower opacity value to voxels on the outer isosurface component than to voxels on the inner ones, we can observe the optically-deeper bubble-spike structures more clearly than in Fig. 2.9(b). Furthermore, by assigning zero opacity to the outermost representative isosurfaces, we can visually extract the involved nested structure only as shown in Fig. 2.9(d). These two visualization results surely reflect the above-mentioned simulation setting.

Case 2: Trailing symmetric isosurface trajectories

Simulated volume datasets such as distributions of energy functions often contain a symmetric isosurface trajectory with respect to the scalar field. In such dataset, the isosurface component at the center value of symmetry occupies the entire volume domain. This type of dataset is often difficult to visualize because we cannot discriminate interesting features such as maxima and minima that invoke the isosurface appearance and disappearance. However, our framework can effectively alleviate this problem using the integral of the isosurface-trajectory distance function, because this topological attribute successfully identifies the center of isosurface trajectory. This leads us to the idea for clearly illuminating the important interior features by lowering the opacity of the voxels belonging to the occluding isosurface.

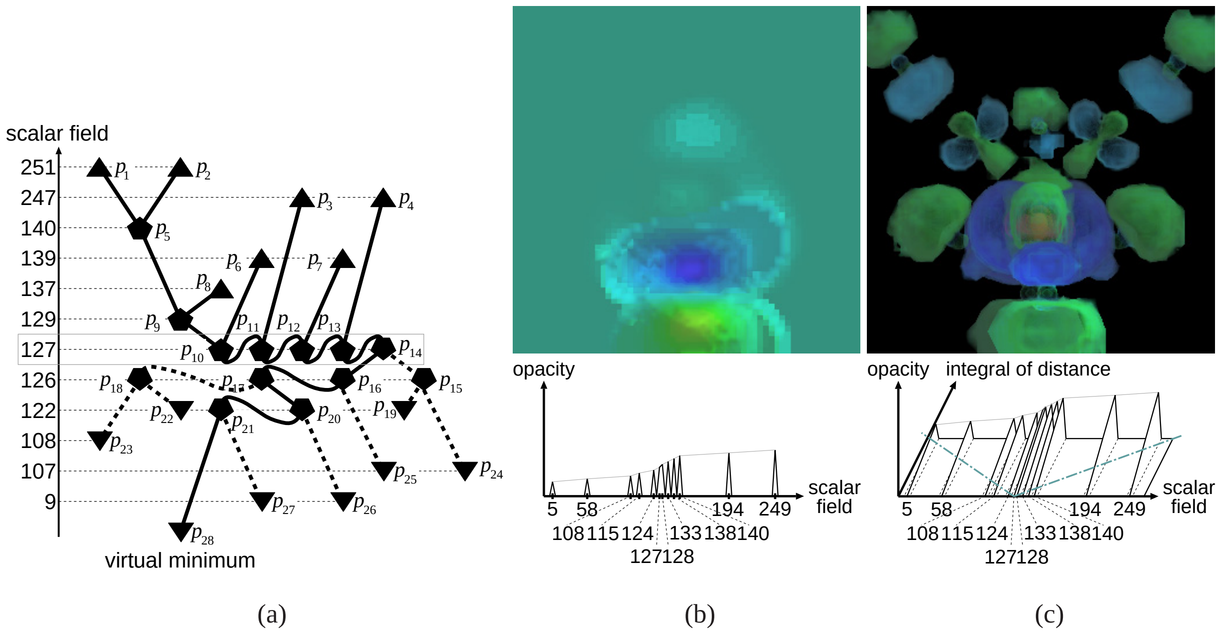


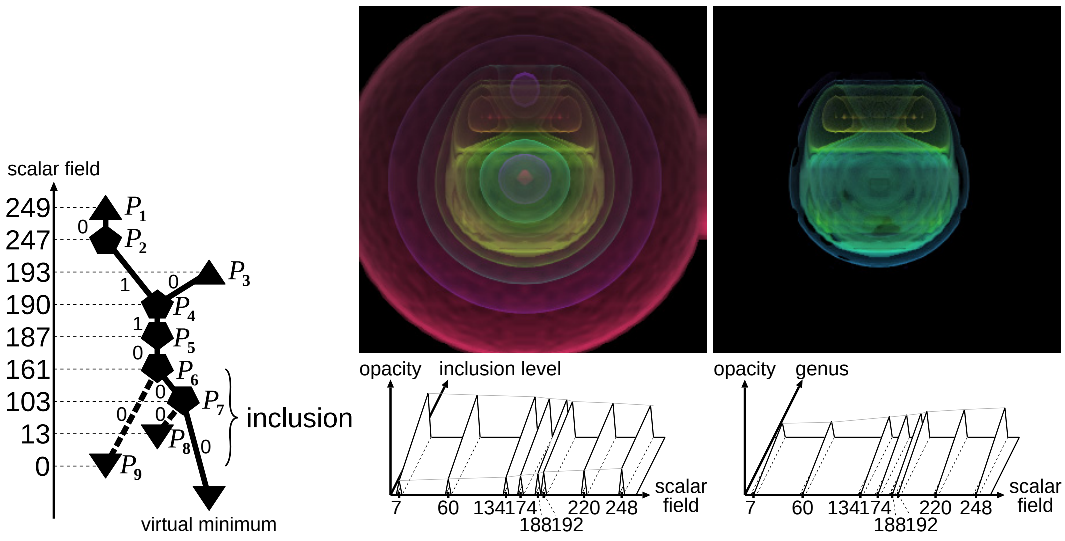
Figure 2.10 Visualizing the HIPIP dataset: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity transfer function, and (c) with 2D opacity transfer function depending also on the integral of the isosurface-trajectory distance function.

For example, as shown in Fig. 2.10, the High Potential Iron Protein (HIPIP) dataset has a symmetric wave function with respect to the scalar field, and thus the isosurface component around the center value of the isosurface trajectory covers up the entire volume. Fig. 2.10(a) shows the VST of the HIPIP dataset, and Fig. 2.10(b) shows a visualization result obtained using the topologically-accentuated 1D opacity transfer function. As seen in Fig. 2.10(a), the VST is almost symmetric and it has many critical points around the mean scalar field value 127. However, the corresponding isosurface component actually occludes many significant features as shown in Fig. 2.10(b) if we assign a large opacity value to each representative field value uniquely. This observation lets us improve the result as shown in Fig. 2.10(c) by lowering the opacity values of the voxels that have the small integral values. Indeed, this allows us to eliminate the occluding isosurface component from the important structures inside the volume. Note that since most voxels have the scalar field values and integral values near the dashed line overlaid in the transfer function definition in Fig. 2.10(c), the opacity value gets substantially higher as the isosurface components shrink with scalar field value less than the mean 127.

Case 3: Accentuating specific isosurface genera

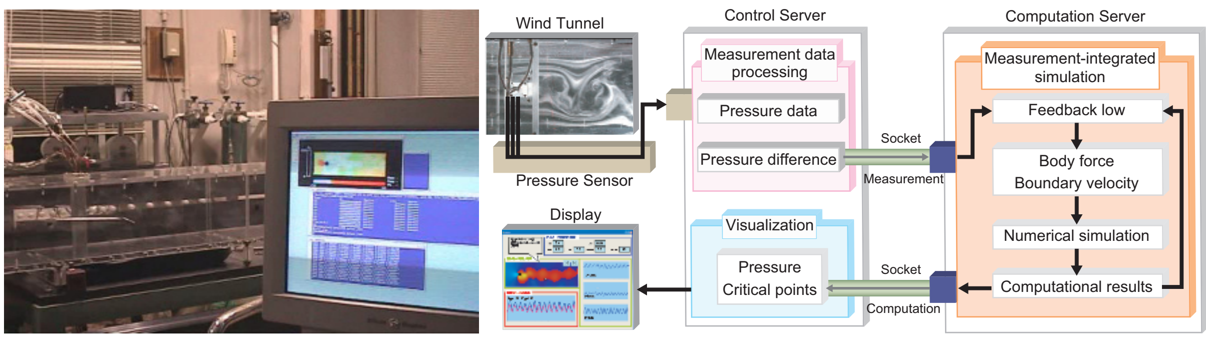
The change in genus of each component of an isosurface may provide an important clue which allows us to visually understand the complexity of the structures embedded in a volume dataset. For example, Fig. 2.11 visualizes the half domain of positive charge distribution simulated around two ^{16}O nucleons [24], where the other nucleon is located above the visualized domain.

Fig. 2.11(a) shows the VST for this dataset, where the number on the left side of each link represents its genus. From the VST, we can see that two isosurface components corresponding to the links $\overline{P_6P_9}$ and $\overline{P_7P_8}$ are included by the outer isosurface component. By assigning higher opacity values to the voxels on the two included isosurface components, we can depict the included isosurface structures in an accentuated manner. The visualization result in Fig. 2.11(b) certainly provides useful information for us to understand the nested structure, though the image cannot be said to provide sufficient information for us to realize the complex interaction between the two nucleons. This motivates us to use the genus of isosurface component instead of the inclusion level as a topological attribute to design the new 2D transfer function that is intended to accentuate the regions topologically equivalent to a torus. A resultant image rendered with the new 2D transfer function is shown in Fig. 2.11(c), where voxels which belong to isosurface components of genus 1 (corresponding to the links $\overline{P_2P_4}$ and $\overline{P_4P_5}$) are emphasized. In fact, the region topologically equivalent to a torus coincides with the subspace having complex interactions between the two nucleons, and attracts much attention from the observers. Furthermore, the visualization result pinpoints the locations where the change in genus is invoked, and provides the observers with important visual cues about the detailed spatial configuration of each of the ^{16}O nucleons.



(a) (b) (c)

Figure 2.11 Visualizing the ^{16}O nucleon dataset: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity transfer function, and (c) with 2D opacity transfer function depending also on the isosurface genus.



(a) Overview (b) System architecture

Figure 2.12 Hybrid wind tunnel system.

2.4.D. A Topologically-Enhanced Juxtaposition Tool for Hybrid Wind Tunnel

In a variety of disciplines, a wide range of measurements and numerical simulations are used to elucidate natural phenomena. Although we can accurately measure physical values such as velocity and pressure at certain points, monitoring an entire region at a given time with the current technology is impossible. On the other hand, although the numerical simulations can capture the physical values of an entire region, it is difficult to emulate real phenomena faithfully. To address this dilemma, measurement-integrated (MI) simulations have been proposed to improve the accuracy of simulations by incorporating measurement data acquired from real phenomena [25].

As an example of MI simulation, we focus on a hybrid wind tunnel system (Fig. 2.12), which allows us to observe 2D Karman vortex streets behind a square cylinder located within the wind tunnel [9]. An accompanying visual analysis tool is required, however, because the resulting flow field changes according to the inlet velocity, which is manually controlled by the users. With such a tool, we can effectively peer into the relationships between actual and simulated flow fields on the fly. Therefore, we proposed a novel visual analysis tool to exploit an augmented reality display [26]. The basic idea behind this tool is to superimpose a computationally-visualized image of the MI simulation onto a corresponding physically-visualized image of the actual flow instantaneously through the course of an experiment. This type of visual analysis tool exemplifies a novel style of visualization called *juxtaposition*, a process proposed by Zabusky et al. in 1993 [27], which involves "*the detailed and quantitative comparison of experimental images with adjacent or superimposed computer simulation images of similar or different functions at the same or different times.*"

The physically-visualized images can be generated simply by a fog generator; the produced oil mist can trace streaklines from the real flow velocity field. On the other hand, a sophisticated means to emphasize the features of the MI-simulated dataset must be designed. Specifically, we employ a differential topology-based scheme to keep track of the Karman vortex street arising in the associated pressure field. Differential topology analysis [28] can extract critical points such as minima, maxima, and saddles, as well as ridge and ravine lines to connect these critical points.

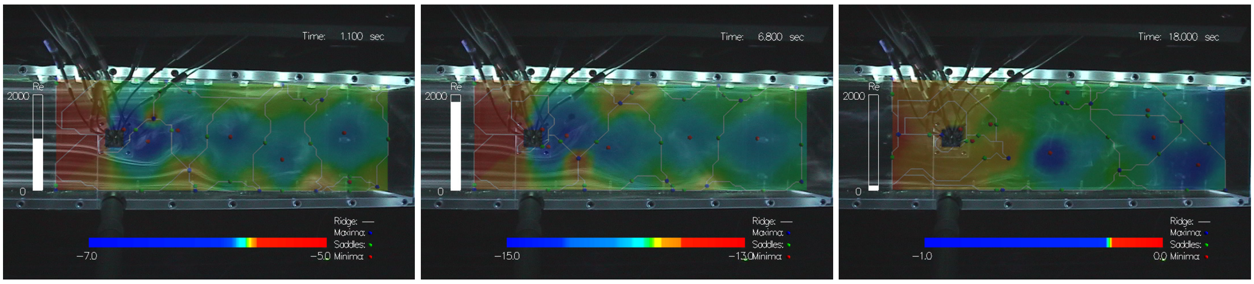


Figure 2.13 Juxtaposition results where the computationally-visualized images are superimposed synchronously onto the corresponding physically-visualized ones.

In flow fields, the pressure value at the center of a vortex is known to give the minimum of the field. The minimum is then surrounded by a ridge cycle, which is constructed by consecutive ridge lines. Therefore, a ridge cycle and its minimum can be regarded as a vortical region and its center, respectively. Keeping this remarkable flow features in mind, our scheme adaptively designs a color map with reference to the field values of critical points to assign more colors to a region in which the distribution of the field values is concentrated. In addition, the ridge cycles are depicted as colored lines bounding the vortex regions. Finally, the image visualized computationally is superimposed onto the one visualized physically. The spatial registration can be performed in a straightforward manner with reference to the square cylinder as a proper marker.

To evaluate the effectiveness of our visual analysis tool, we applied it to an actual dataset form the hybrid wind tunnel system. We superimposed our computationally-visualized results synchronously onto the physically-visualized images of the wind tunnel, as shown in Fig. 2.13. The oil mist in the physically-visualized images allows us to trace the physical streaklines, whereas on the computationally-visualized image layer for the MI-simulated dataset, the maxima, saddles, and minima are plotted in blue, green, and red, respectively, and the ridge lines are depicted in white. Spatial registration of these two visual sources was performed easily with reference to the position of the square cylinder. We can obtain not only streaklines but also the pressure field distribution and the centers of vortices simultaneously from these juxtaposed images, with which we can effectively peer into the relationships between the actual and simulated flow fields. Such overlaid juxtaposition made it possible to visually convey the information which cannot be obtained solely with computational images nor physical images.

2.5. Conclusion

In this section, we introduced the three main themes of our study. The first is the cooperative visualization environment called TFI-AS/V (Transdisciplinary Fluid Integration-Archive System/Visualization), which not only assists users in selecting the best visualization techniques but also provides them with support for browsing and maintaining detailed provenance information throughout the visualization lifecycle. The second is the time-critical rendering scheme for particle visualization, which renders as many accurate particles as possible while maintaining an interactive frame rate. The last is the advanced visualization schemes based on topological structure.

In addition, we proposed a 6DOF haptic transfer function design method for effective exploration of 3D diffusion tensor fields, texture-based emotional visualization of musical performances, and visual simulation of 3D flows on the basis of 2D numerical solutions.

Acknowledgments

The authors would like to thank Prof. Hitoshi Sakagami at University of Hyogo to let us use the implosion datasets in Section 2.4 and provide many valuable comments.

References

[1]Takeshima, Y., Fujishiro, I., and Hayase, T., “GADGET/FV: Ontology-Supported Design of Visualization Workflows in Fluid Science,” *DVD Proceedings of The First International Workshop on Super Visualization*, June 2008.

[2]Cameron, G., “Special Focus: Modular Visualization Environments (MVEs),” *Computer Graphics*, Vol. 29, No. 2, May 1995, pp. 3–60.

[3]Fujishiro, I., Takeshima, Y., Ichikawa, Y., and Nakamura, K., “GADGET: Goal-Oriented Application Design Guidance for Modular Visualization Environments,” *Proceedings of IEEE Visualization ’97*, 1997, pp. 245–252, 548.

[4]Fujishiro, I., Furuhata, R., Ichikawa, Y., and Takeshima, Y., “GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment,” *Proceedings of IEEE Symposium on Information Visualization 2000*, 2000, pp. 77–83.

[5]Takeshima, Y. and Fujishiro, I., “A Design Support Environment for Flow Visualization Applications,” *Proceedings of Second International*

Symposium on Transdisciplinary Fluid Integration, 2005, p. 48.

[6]Wehrend, S. and Lewis, C., "A Problem-Oriented Classification of Visualization Techniques," *Proceedings of IEEE Visualization '90*, 1990, pp. 139–143.

[7]Johnson, C. R., Moorhead, R., Munzner, T., Pfister, H., Rheingans, P., and Yoo, T. S., editors, *NIH-NSF Visualization Research Challenges Report*, IEEE Computer Society Press, 2006.

[8]Keller, P. R. and Keller, M. M., *Visual Cues – Practical Data Visualization*, IEEE Computer Society Press, 1993.

[9]Hayase, T., Nisugi, K., and Shirai, A., "Numerical Realization for Analysis of Real Flows by Integrating Computation and Measurement," *International Journal for Numerical Methods in Fluids*, Vol. 47, 2005, pp. 543–559.

[10]Fujishiro, I., Takeshima, Y., and Hayase, T., "Prototyping of Cooperative Visualization Environment and Its Application to Transdisplinary Fluid Integration Research," *Proceedings of JSME Annual Congress 2007*, 2007, pp. 175–176, (in Japanese).

[11]Fujishjiro, I., Takahashi, S., and Takeshima, Y., "Collaborative visualization: Topological approaches to parameter tweaking for informative volume rendering," *Systems Modeling and Simulation: Theory and Applications, Asia Simulation Conference 2006*, Springer-Verlag, 2006, pp. 1–5.

[12]Takeshima, Y., Fujishiro, I., and Tokumasu, T., "A PID Control-Based Scheme for Time-Critical Rendring of Particle Systems," *Poster Proceedings of IEEE Pacific Visualization 2011*, 2011.

[13]Gribble, C. P. and Parker, S. G., "Enhancing interactive particle visualization with advanced shading models," *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, 2006, pp. 111–118.

[14]Ellsworth, D., Green, B., and Moran, P., "Interactive Terascale Particle Visualization," *Proceedings of the Conference on Visualization 2004*, 2004, pp. 353–360.

[15]Gumhold, S., "Splatting Illuminated Ellipsoids with Depth Correction," *Proceedings of the Vision, Modeling, and Visualization Conference 2003*, 2003, pp. 245–252.

[16]Tokumasu, T., Kamijo, K., Oike, M., and Matsumoto, Y., "Molecular Dynamics Study of the Nucleation of Bubble," *Proceedings of Fourth International Symposium on Caviation*, 2001, pp. A1–005.

[17]Takahashi, S., Nielson, G. M., Takeshima, Y., and Fujishiro, I., "Topological Volume Skeletonization Using Adaptive Tetrahedralization," *Proceedings of Geometric Modeling and Processing 2004*, 2004, pp. 227–236.

[18]Takahashi, S., Takeshima, Y., and Fujishiro, I., "Topological Volume Skeletonization and Its Application to Transfer Function Design," *Graphical Models*, Vol. 66, No. 1, 2004, pp. 24–49.

[19]Bajaj, C. L., Pascucci, V., and Schikore, D. R., "The Contour Spectrum," *Proceedings of IEEE Visualization '97*, 1997, pp. 167–173.

[20]Takahashi, S., Fujishiro, I., Takeshima, Y., and Nishita, T., "A Feature-Driven Approach to Locating Optimal Viewpoints for Volume Visualization," *Proceedings of IEEE Visualization 2005*, 2005, pp. 495–502.

[21]Naraoka, R., Fujishiro, I., Takahashi, S., and Takeshima, Y., "Locating an Optimal Light Source for Volume Rendering," *DVD Proceedings of IIEEJ Image Electronics and Visual Computing Workshop 2007*, 2007, pp. 4B–2.

[22]Takeshima, Y., Takahashi, S., Fujishiro, I., and Nielson, G. M., "Introducing Topological Attributes for Objective-Based Visualization of Simulated Datasets," *Proceedings of International Workshop on Volume Graphics*, 2005, pp. 137–236.

[23]Sakagami, H., Murai, H., Seo, Y., and Yokokawa, M., "14.9 TFLOPS Three-dimensional Fluid Simulation for Fusion Science with HPF on the Earth Simulator," *Proceedings of Supercomputing 2002*, 2002.

[24]Meißner, M., "`[http://www.volvis.org/]`,".

[25]Hayase, T. and Hayashi, S., "State Estimator of Flow as an Integrated Computational Method with the Feed-Back of Online Experimental Measurement," *ASME Journal of Fluids Engineering*, Vol. 119, No. 4, 1997, pp. 814–822.

[26]Takeshima, Y., Fujishiro, I., Takahashi, S., and Hayase, T., "A Topologically-Enhanced Juxtaposition Tool for Hybrid Wind Tunnel," *Proceedings of IEEE Pacific Visualization 2013*, Feb. 2013.

[27]Zabusky, N. J., Silber, D., Pelz, R., and Vizgroup'93, "Visiometrics, Juxtaposition and Modeling," *Physics Today*, Vol. 46, No. 3, 1993, pp. 24–31.

[28]Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T. L., and Ueda, M., "Algorithms for Extracting Correct Critical Points and Constructing Topological Graphs from Discrete Geographical Elevation Data," *Computer Graphics Forum*, Vol. 14, No. 3, 1995, pp. 181–192.