J Vis
DOI 10.1007/s12650-013-0180-3

REGULAR PAPER

Jianlong Zhou · Chun Xiao · Masahiro Takatsuka

A multi-dimensional importance metric for contour tree simplification

Received: 1 November 2011 / Revised: 16 March 2013 / Accepted: 5 July 2013

Abstract Real-world data sets produce unmanageably large contour trees because of noise. Contour Tree Simplification (CTS) would remove small scale branches, and maintain essential structure of data. Despite multiple measures of importance (MOIs) available, conventional CTS approaches often use a single MOI, which is not enough in evaluating the importance of branches in the CTS. This paper proposes an importance-driven CTS approach. The proposed approach combines multiple MOIs through the introduction of various concepts to maximize advantages of each MOI. In the attribute space, various attributes of a branch are organized in a single space. The concept of the importance triangle is used to evaluate the importance of a branch by size of the importance triangle. It considers the whole attribute space and gives better evaluation of importance. Finally, new importance values of branches are compared in the importance space to make simplification decisions.

Keywords Contour tree simplification · Attribute space · Multi-dimensional · Importance-driven · Measure of importance · Volume rendering

1 Introduction

Topology has been an important tool for analyzing scalar data and flow fields in visualization. Topological features of a field are characterized by its critical points. Two data structures are commonly used for explicitly storing topological features: Morse-Smale (MS) complexes (Edelsbrunner 2001) and (Reeb 1946) graphs. The MS complex decomposes domain of a function into regions having uniform gradient flow (Smale 1961). The Reeb graph (Reeb 1946) is a simple structure that summarizes the topology of a Morse function. It traces components of isosurfaces/contours as they sweep the domain. For functions with simply connected domains, this graph is also simply connected and called the contour tree (CT). This paper focuses on the contour tree for storing topological features. The concept of isosurface/contour is used to set up the

J. Zhou
The University of Sydney, Sydney, Australia

J. Zhou (✉)
National ICT Australia (NICTA), Sydney, Australia
E-mail: jianlong.zhou@nicta.com.au
Tel.: +61-2-93762223

C. Xiao
College of Information Engineering, Xiangtan University, Xiangtan, Hunan, China

M. Takatsuka
School of Information Technologies, The University of Sydney, Sydney, Australia

Published online: 24 August 2013

contour tree. The isosurface is defined as follows: Given a field $f: R^3 \rightarrow R$, the isosurface of f for an isovalue h is the inverse image $f^{-1}(h)$ of the isovalue. The contour tree traces components of isosurfaces as they sweep the domain. It represents the nesting relationships of connected components of isosurfaces (Carr et al. 2010; Weber et al. 2007). Typically, the contour tree is represented as a list of nodes and a list of arcs. Pascucci (2004) used an alternative branch decomposition where a branch is defined as a monotone path in the graph traversing a sequence of nodes with non-decreasing (or non-increasing) value of the scalar field.

The contour tree is vulnerable to noise which adds small scale topological features and causes the contour tree size to increase dramatically (Carr et al. 2010). This makes it difficult to recognize branches that correspond to objects of interest, and results in the contour tree being impractical in data analysis and visualization. As a result, Contour Tree Simplification (CTS) would remove unimportant branches, while making the size of the tree small enough for the user interaction and maintaining essential structure of the data. However, despite multiple MOIs available (Carr et al. 2010), conventional approaches still have the following problems: (1) they often use single measure to evaluate importance of branches. Because various measures of importance emphasize different features of data sets, it is obvious that single MOI is not enough in evaluating importance of branches; (2) to determine a region that preserved branches located in the attribute space, a user has to estimate multiple linear discriminating functions.

In this paper, we propose an importance-driven approach for the CTS. The proposed approach uses multiple measures through introducing concepts of attribute space, importance triangle, and importance space into the CTS pipeline. The objective of this work is to deliver a new paradigm of the CTS for making full use of advantages of multiple MOIs simultaneously and improve the CTS efficiency. The proposed approach can be generalized to process branches with more than three MOIs. The contributions of the paper are as follows:

- A concept of attribute space is proposed to organize various attributes of a branch in a single space. As a result, the importance of various branches can be compared in an importance space during the CTS.
- A concept of importance triangle is presented to evaluate the importance of branches, which can take full advantages of multiple measures simultaneously.
- A single simplification threshold considers multiple MOIs simultaneously and allows users to manipulate thresholds more meaningfully and efficiently.

2 Related work

The topology simplification process suppresses insignificant features by removing or canceling pairs of critical points that are considered unimportant in terms of a specified measure. Carr et al. (2010) simplified the contour tree with two basic operations: leaf pruning and node reduction. Leaf pruning removes topology from the field by selecting a leaf node of low importance and removing it. While node reduction eliminates connectivity regular points from the contour tree, leaving the topology unchanged. The scheme involves computation of several measures that are used for ranking the "importance" of an arc. The measures used in the CTS include persistence, volume and hyper volume. However, these measures are used individually in the CTS process.

Pascucci et al. (2004) presented a multi-resolution data structure—branch decomposition—to represent contour trees. A priority queue is used to store leaf branches of join tree and split tree during construction of the contour tree. The priority for each branch in the scheme is the persistence. Pascucci et al. (2007) also used persistence based simplification to eliminate insignificant saddle-extremum pairs from the Reeb graph. Takahashi et al. (2004b) simplified the contour tree using persistence of various node patterns. The number of simplification steps is controlled by a threshold. Besides contour trees, MS complexes are used to store the topological information of a 3D data set (Smale 1961; Natarajan and Pascucci 2005). The persistence is used by (Edelsbrunner et al. 2001) for the simplification of MS complexes. It is defined to be the number of steps of this sweep for which a feature retains its topological uniqueness.

In a word, the previous work on the topology simplification often used various MOIs separately. Because each MOI has its specific emphasis on data features, it is obvious that previous work did not take full advantages of each measure. We strongly believe that appropriate integration of various MOIs will produce a better topology simplification. This paper focuses on effective evaluation on the importance of branches in the CTS.

3 Measures of importance

In this paper, we use a priority queue to keep track of branches of the tree with their associated priorities. The priority of each branch is equal to its importance value. We also use the branch decomposition (Pascucci 2004) in the CTS pipeline.

3.1 Definition of importance

In the (Merriam-Webster 2010) dictionary, importance is defined as: *Importance means a quality or aspect having great worth or significance. It implies a value judgment of the superior worth or influence of something or someone*. It describes the quality (positive or negative) that renders something desirable or valuable, and worthy of note.

In this paper, we define importance as follows: Importance suggests an evaluation or judgment of significance of an object in a data set. It describes the quality that renders an object desirable or valuable, and worthy of note in visualization. This quality is represented by some measures that evaluate the degree of an object which draws attention to viewers in visualization. In the CTS, importance is a simplification value that indicates the branch's significance. Branches with lower importance are candidates to be removed during the CTS.

3.2 Evaluation of importance

In the contour tree, each branch corresponds to a region in the data domain. The importance of each region can be depicted using different measures. These measures are then used to drive the CTS process. The importance of one object is related to different features of the scalar field, e.g. scalar value, size, position, and their combinations. The persistence p, volume v and hyper volume hv belong to measures derived from the data set itself. Persistence is equal to the absolute difference in scalar values of two critical points. Volume is the voxel count of the region enclosed by the isosurface. Hyper volume is the integral of the scalar field over the enclosed region.

From the physical point of view, persistence, volume and hyper volume describe the importance of a branch from different physical aspects. For example, when we think the scalar value of each voxel as the mass of that voxel, the importance described by hyper volume is based on the mass of the region corresponding to a branch, i.e. what the weight of a branch is. While persistence describes importance based on the number of steps of the sweep for which a feature retains its topological uniqueness, and volume describes the importance based on the size of the region corresponding to a branch.

From the importance's point of view, these measures are different descriptors of importance for a branch. For example, giving two branches $b1$ and $b2$ with same p and v, $f\max = 100$, $f\min = 2$, $v1 = v2 = 50$, $p1 = p2 = 100-2 = 98$. There are 49 voxels whose scalar value is 100 and 1 voxel whose scalar value is 2 in $b1$, while there are 49 voxels whose scalar value is 2 and 1 voxel whose scalar value is 100 in $b2$. In this case, we get $hv1 = 4{,}902$, $hv2 = 198$. From this example, we see that persistence and volume cannot decide hyper volume uniquely. Hyper volume is an independent MOI from the importance's point of view. So we treat persistence, volume, and hyper volume as three different MOIs, and make full use of advantages of each measure in our approach in the CTS pipeline.

Meaningful and important features are not always captured by the notion of persistence. This is also true for volume and hyper volume. Persistence has the advantage of highlighting high scalar range regions, but it easily suppresses large objects with limited ranges of voxel intensity. Volume has the power to easily preserve objects with large spatial extent, but it easily suppresses small spatial extent objects. Hyper volume may regard high-intensity noise or artifacts as objects of interest. Each MOI does not on its own describe all relevant features. Various measures need to be combined together to more accurately depict importance of objects. The contributions of different MOIs for the final importance value I are different, and can be expressed as: $I = g\ (p,\ v,\ hv)$.

4 Importance-driven contour tree simplification

This section presents an importance-driven approach, which combines multiple MOIs to take their full advantages in the CTS pipeline. In this paper, the concepts of attribute space, importance triangle, and

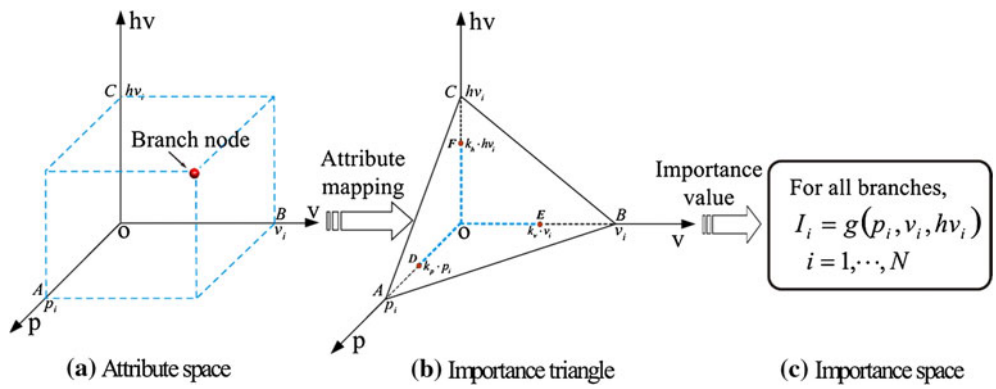


Fig. 1 The pipeline of importance evaluation in the importance-driven contour tree simplification

importance space are introduced into the CTS pipeline. Figure 1 shows the process of importance evaluation in the importance-driven contour tree simplification. Various measures are firstly represented in the attribute space. Then the importance triangle is used to map multiple MOIs onto one value. The new importance values of branches are compared in the importance space to make simplification decisions. The details of each part will be covered in the later subsections.

4.1 Attribute space

We consider three MOIs in the CTS: persistence, volume and hyper volume. The goal of our approach is to combine three MOIs to evaluate the importance of a branch, trying to keep advantages and minimize disadvantages of each measure during the CTS. So this problem can be expressed as follows: suppose that we have N branches in the contour tree, each branch has a property field which stores a 3-dimensional vector representing three importance measures of p, v, and hv. These N vectors are represented as Mi, $i = 1,\ldots, N$. We need to find a mapping which maps a 3-dimensional vector Mi of importance measures onto a scalar value. The importance measure vector is represented as: $\mathrm{M}i = [pi\ vi\ hvi]^{\mathrm{T}}$.

In order to allow to use MOIs with various units together in a single pipeline, this paper introduces an abstract space—attribute space—into the CTS pipeline. In the attribute space, each MOI is represented with an axis in the 3D Cartesian coordinate system as shown in Fig. 1a. Importance values on each axis are represented with a single unit—relative importance. The relative importance is used to show the importance of one branch relative to the branch with the peak importance value in the contour tree. In this way, although each MOI has different unit, the abstract level of relative importance evaluates different MOIs using a single unit. So the relative importance can be used to evaluate the combination of various MOIs in one space. In the attribute space, a node is used to represent a branch with specific persistence, volume and hyper volume as shown in Fig. 1a.

4.2 Importance triangle

As mentioned, we need to evaluate a new importance value for each branch based on components of M_i. We represent M_i in the attribute space as shown in Fig. 1b, where the coordinates of points A, B and C are (p_i, 0, 0), (0, v_i, 0) and (0, 0, hv_i) respectively. A triangle ΔABC is then set up to represent the vector M_i. In this way, each vector M_i is mapped to a unique triangle ΔABC. This triangle is named as importance triangle (*ITri*).

In the attribute space, three MOIs form a tetrahedra *OABC*. The tetrahedra represents the overall contribution of multiple MOIs for the importance of a branch. This motivates us to get the importance evaluation of a branch based on multiple MOIs by evaluating properties of the tetrahedra. In the tetrahedra, every two edges (e.g. *OA*, *OB*, and *OC* in Fig. 1b) form a triangle. Each triangle represents the importance contributed by two corresponding MOIs of the triangle in the attribute space. Inspired by (Takahashi et al. 2004a) approach of using the product between volume and persistence as a weight for the CTS, this paper uses the product of each pair of MOIs as the principal importance factor in the evaluation of importance of the multiple MOIs. Each product corresponds to the double size (area) of the corresponding triangle of the

tetrahedra. Because the area of triangles in the tetrahedron *OABC* has the relation as shown in Eq. 1 (Fitting 2001; Quadrat et al. 2001), we use the area of the triangle ΔABC as the final value of relative importance in the CTS:

$$S^2_{\Delta ABC} = S^2_{\Delta OAB} + S^2_{\Delta OBC} + S^2_{\Delta OAC} \tag{1}$$

where S is the area of various triangles. The size (area) of *ITri* is computed with Eq. 2:

$$S_i = \frac{1}{2}\left\|\overrightarrow{AB} \times \overrightarrow{AC}\right\| = \frac{1}{2}\sqrt{(hv_i \cdot p_i)^2 + (v_i \cdot p_i)^2 + (hv_i \cdot v_i)^2}, \tag{2}$$

where S_i is the area of *ITri* of the ith branch. The final importance value of the ith branch I_i is defined to be equal to S_i ($I_i = S_i$). Because I_i is based on the size of *ITri*, we call this measure of importance as *ITri*. As mentioned, branches with small p_i, v_i and hv_i correspond to noise physically in the data space, where S_i is also small according to Eq. 2. Therefore, *ITri* provides a method to evaluate importance of branches based on multiple MOIs. Physically, it focuses on applying small importance values to branches with small p_i, v_i and hv_i, which correspond to noise and will be removed during the CTS.

Furthermore, the products used to evaluate size of triangles of the tetrahedra have obvious physical meanings. For example, the product of volume and persistence represents the size of the 4D subspace swept by the corresponding isosurface, which is contained in the entire 4D space spanned by the (x, y, z)-coordinates and scalar field (Takahashi et al. 2004a). Therefore, the size of the *ITri* is a physically meaningful measure of importance which combines multiple MOIs in the pipeline.

In general, if the ith branch has n importance measures m_{ij} ($j = 1,\ldots, n$), the importance measure vector becomes an n-dimensional vector. In this case, the area of *ITri* used in the case of 3-dimensional measure vector is extended to the concept of the area of the *hypotenuse face* (Quadrat et al. 2001) in the case of an n-dimensional vector. The area of the hypotenuse face is used to evaluate contribution of all measure variables m_{ij} ($j = 1,\ldots, n$) of the ith branch to the final importance value and computed with Eq. 3:

$$I_i = \frac{1}{n-1}\sqrt{\sum_{k=1}^{n}\prod_{j=1, j\neq k}^{n} m_{ij}^2}, \tag{3}$$

After getting the importance value I_i of each branch, importance values of all branches based on the concept of importance area are compared in one space—the importance space. The concept of importance space organizes all branches in one space, and allows compare importance of branches based on multiple measures of importance. In the importance space, a threshold I_t is used to control the simplification level: a branch that has larger importance value of I_i than the specified threshold I_t is removed during the CTS.

4.3 Advantages

This subsection compares *ITri* with conventional approaches in three aspects: importance value, regions of preserved branches, and threshold in the importance space.

In the tetrahedron formed from various MOIs as shown in Fig. 1b, if a conventional method, such as the single measure, is used (Carr et al. 2010), the importance value corresponds to the length of the line of *OA*, *OB* or *OC* in the attribute space. If weighted summation of various MOIs (as shown in Eq. 4) is used to get the importance value in the CTS, the final importance value corresponds to the summation of the length of three lines *OD*, *OE* and *OF* in the attribute space as shown in Fig. 1b.

$$I_i = k_p \cdot p_i + k_v \cdot v_i + k_h \cdot hv_i, \tag{4}$$

where k_p, k_v and k_h are weighting coefficients. It is obvious that these conventional methods only consider part of the attribute space in the CTS pipeline. Compared with conventional methods, the final importance value of our approach corresponds to area of triangles instead of the length of lines. It considers the whole attribute space and gives better evaluation on the importance of a branch during the CTS.

Figure 2a is a 2D diagram to show the comparison of *ITri* and persistence as MOIs in the CTS. In this figure, the red curve represents the threshold I_t where *ITri* is used as the MOI in the CTS, and the vertical blue line represents the threshold p_t where persistence p is used as MOI in the CTS. When *ITri* is used in the CTS, branch nodes positioned on the top side of the red curve (i.e., the region C and D) are preserved. By contrast, if persistence is used as a single MOI in the CTS, branch nodes positioned on the right side of the blue line (i.e., the region A and D) are preserved in the CTS. The difference between *ITri* and persistence is

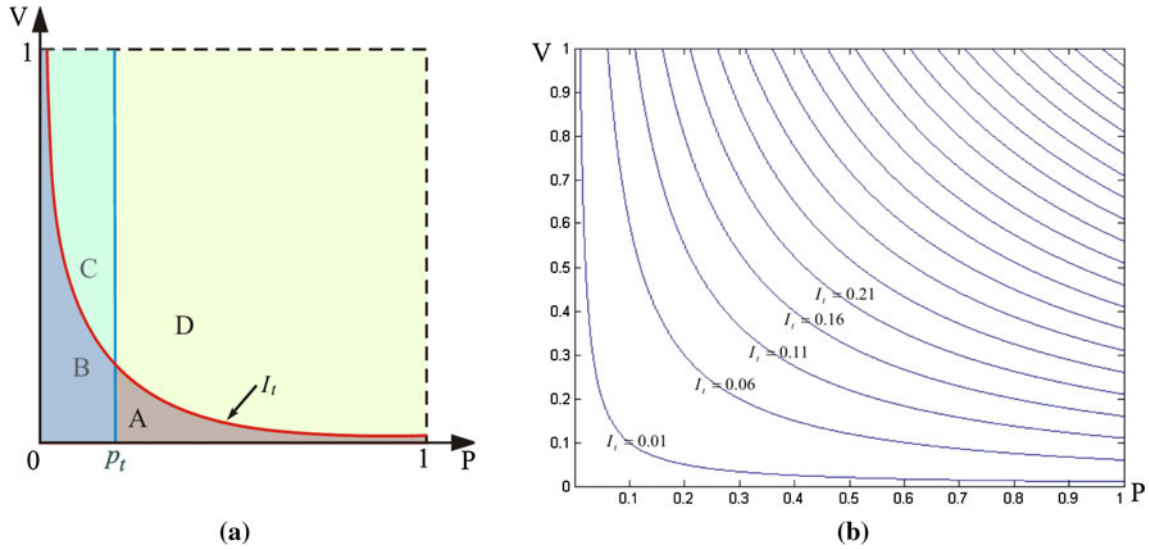


Fig. 2 Comparison of *ITri* and Conventional Approaches: (**a**) The comparison of *ITri* and persistence; (**b**) A set of threshold curves of *It* with different *It* in a two-dimensional case

obvious: persistence removes branches in the region C which are considered as branches of interest by *ITri,* while preserves branches in the region A which are considered as noise by *ITri*. Despite persistence of branches in the region C being smaller than p_t, they are possibly branches of interest considering their corresponding volume and hyper volume. Similarly, despite persistence of branches in the region A being larger than p_t, they are possibly noise considering their corresponding volume and hyper volume. *ITri* balances these considerations and simplify the contour tree more effectively. The difference between *ITri* and volume or hyper volume in the CTS is similar with that between *ITri* and persistence.

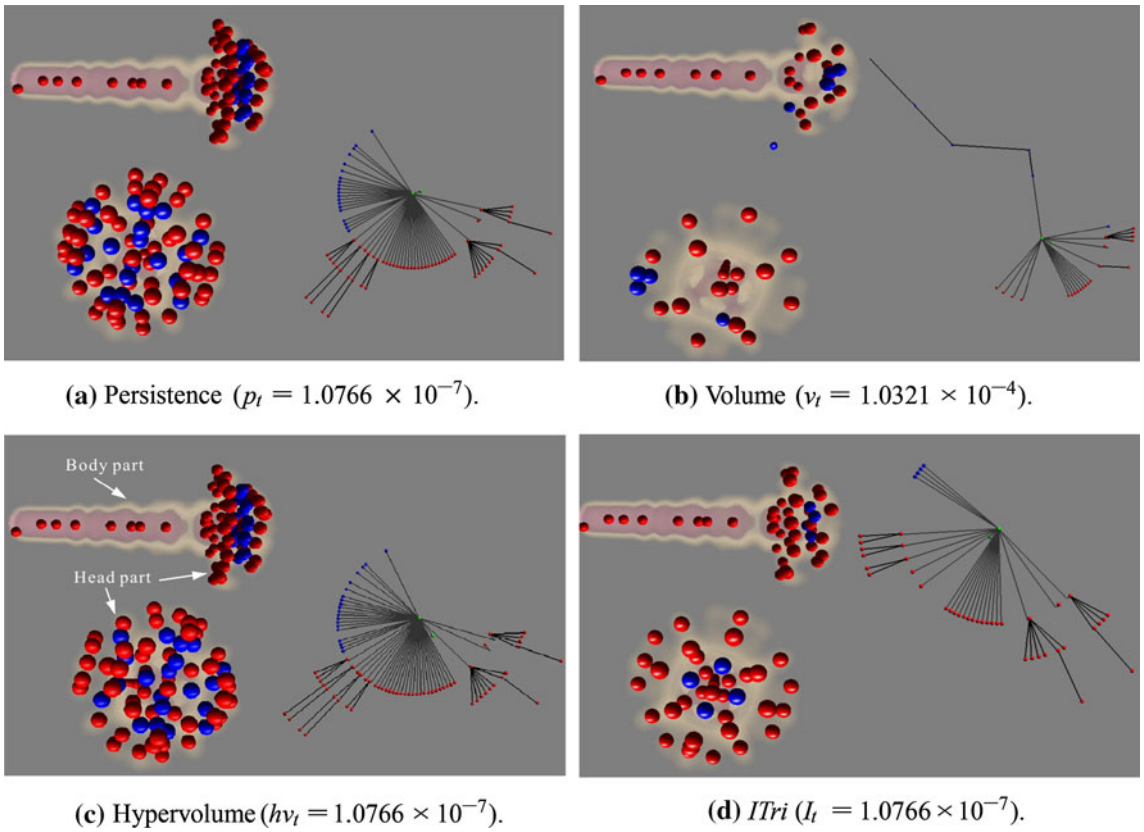
Figure 2b shows a set of threshold curves with different I_t in the importance space in a 2D diagram. I_t is increased from bottom left to top right in this figure. From these curves we see that persistence, volume and hyper volume of preserved branches are increased accordingly with the increase of I_t (we only show persistence and volume in Fig. 2b in this 2D example). By contrast, if a single measure such as persistence is used, only persistence of preserved branches is increased with the increase of p_t. Even if other measures (e.g. volume) are zero, branches are still preserved as long as their persistence is larger than p_t.

From Fig. 2a, we see that if branches in the region C and D need to be preserved based on conventional approaches, users have to determine multiple discriminating functions (i.e. multiple thresholds). One threshold of volume needs to be specified besides the threshold of p_t to approximately determine the region C and D. The determination of multiple thresholds for a given task is often difficult. This determination process lacks guiding information and is often based on trial-and-error process. On the contrary, *ITri* only needs one threshold to determine the region C. It balances contributions of multiple measures for the final importance value.

5 Results and discussions

We conducted experiments on various data sets to demonstrate the effectiveness and utility of the proposed approach. Our system was run on Windows XP platform on a Dell machine equipped with 3GiB RAM and an NVIDIA GeForce 8300GS graphics card.

Figure 3 shows the rendering and topology of the "fuel" data set (see http://www.volvis.org/). The data set is rendered using volume rendering with the critical points drawn on their original positions, which helps users to understand the data topology visually (red nodes represent the local maximum points, blue nodes represent the local minimum points, and green nodes represent the root points in this paper). The corresponding level of the simplified contour tree graph is drawn on the right hand side of the rendered data set. The contour tree graphs in this paper are drawn based on the Orrery-like arrangement (Pascucci 2004). p_t, v_t, hv_t and I_t are thresholds used in the CTS. In this experiment, we set p_t, hv_t and I_t be equal in order to compare effectiveness of various measures. From the comparison, we see that Fig. 3a, c are similar, but



(a) Persistence ($p_t = 1.0766 \times 10^{-7}$).

(b) Volume ($v_t = 1.0321 \times 10^{-4}$).

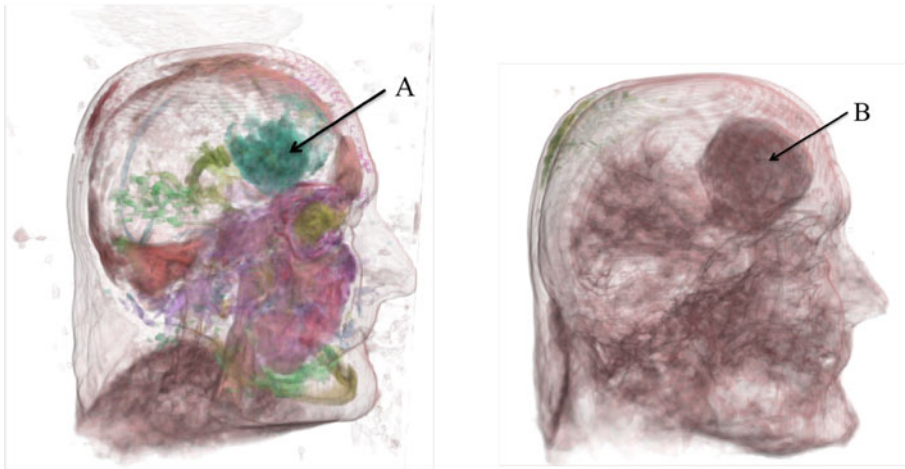
(c) Hypervolume ($hv_t = 1.0766 \times 10^{-7}$).

(d) *ITri* ($I_t = 1.0766 \times 10^{-7}$).

Fig. 3 Rendering and topology of "fuel" data set. The object on the *lower left side* in each image is the *right view* of the object on the *upper left side*, and the corresponding contour tree is on the *right hand side*

more branches are preserved in Fig. 3a. This is because that hyper volume considers the size of features besides the scalar value during the evaluation of importance of features. From the hyper volume's point of view, a branch will be removed if its hyper volume is small even if its persistence is large. In Fig. 3b, although volume can preserve some critical points successfully, it cannot remove noise (e.g. the blue point outside the structure) effectively even if the threshold is larger than that of other measures. In Fig. 3c, d, we see that the simplification results of two methods are same at the body part because topological subregions are relatively larger than that at the head of the data, and hyper volume can capture this feature as *ITri* does because of larger hyper volume. However, it is clear that the results are completely different at the head part of the data. Hyper volume cannot capture features at the head part of the object, while *ITri* can capture these evenly circularly distributed topological subregions successfully.

For a domain specific case, for example, in a medical data set with tumors inside, tumors often have low ranges of scalar values relative to surrounding objects. Furthermore, there are also different sizes of tumors. In order to simplify the contour tree of this kind of data set, users need to preserve branches with small persistence while considering volume and hyper volume at the same time. Figure 4 presents the rendering of the "TumorHead" data set (data courtesy of B Terwey, Bremen), where there is a brain tumor pointed out by A in Fig. 4a. Given the number of preserved branches being same for persistence and *ITri* after the simplification, the simplified contour tree is used to generate transfer functions using the approach presented in (Zhou and Takatsuka 2009). This experiment aims to compare differences of rendering results using transfer functions based on CTS measures of persistence and *ITri*. In this experiment, the goal of the CTS is to preserve the branch corresponding to the tumor of A as shown in Fig. 4a and the tumor B as shown in Fig. 4b. From the comparison of Fig. 4a, b, we see that both images preserve the tumor structure. However, differences are obvious: the size of tumor rendered based on persistence is obviously smaller than that of tumor rendered based on *ITri*. This is because that the persistence of the tumor is small, some of branches corresponding to the tumor are removed when persistence is used as the MOI during the CTS.



(a) Persistence. **(b)** *ITri*.

Fig. 4 Rendering of "TumorHead" data set using transfer functions generated from contour tree with various CTS measures

From the experiments, we see that the proposed approach can simplify the contour tree meaningfully and effectively. The utility of our approach is two-fold: as a comprehensive solution for combining multiple MOIs in the CTS, and as a general effective interface to manipulate thresholds in the CTS. Our approach provided a complete pipeline for users to organize, display, and simplify the contour tree. It showed advantages in the CTS process. Another advantage of our approach is that it can be easily extended as a general scheme to simplify the contour tree with more than three MOIs.

The proposed approach has wide applications in visualization. For example, by using the proposed approach, the subregions in a data set can be precisely indexed by branches in the contour tree. This is one of crucial steps in improving effectiveness of topology controlled automation of transfer function generations in volume rendering (Zhou and Takatsuka 2009). The proposed approach can also be used in visual representation of topological structures of data sets, such as topological spines (Correa et al. 2011) like structure of a data set in visual analysis. The simplified contour tree can be used in isosurface extractions, and guiding exploratory visualization (Carr et al. 2010). It can also be used in the automated image segmentation (Johansson 2007).

6 Conclusions and future work

This paper presented an importance-driven approach for the CTS. The proposed approach combined multiple MOIs into a CTS pipeline through the introduction of various concepts. The presented approach demonstrated advantages in the use of MOIs. We presented three measures in this paper. More MOIs can be developed to represent features of data sets and combined with the proposed approach. Another possible direction for the future work is to extend the proposed approach and use it as a general framework to combine multiple features of data sets in other applications. As a future work, an objective criterion will be developed to evaluate the proposed approach. For example, a data set can be segmented using a segmentation method. The difference (of volume, hyper volume, etc.) between the subregion indexed by branch and the corresponding subregion from the segmentation can be measured to show the effectiveness of the CTS approach.

References

Carr H, Snoeyink J, van de Panne M (2010) Flexible isosurfaces: simplifying and displaying scalar topology using the contour tree. Comp Geo 43(1):42–58

Correa C, Lindstrom P, Bremer P-T (2011) Topological Spines: a structure-preserving visual representation of scalar fields. IEEE Trans Visual Comput Graphics 17(12):1842–1851

Edelsbrunner H, Harer J, Zomorodian A (2001) Hierarchical morse complexes for piecewise linear 2-manifolds. In Proceedings of the 17th annual symposium on Computational geometry, pp 70–79
Fitting M (2001) Pythagoras' Theorem for areas—revisited. Technical report, department of mathematics and computer science, City University of New York
Johansson G, Museth K, Carr H (2007) Flexible and topologically localized segmentation. In Proceedings of Eurographics/IEEE VGTC Symposium on Visualization (EuroVis2007) pp 179–186
Merriam-Webster (2010). Merriam-Webster Online. http://www.merriam-webster.com/dictionary/importance
Natarajan V, Pascucci V (2005) Volumetric data analysis using morse- smale complexes. In: Proceedings of the International Conference on Shape Modeling and Applications 2005 pp 322–327
Pascucci V, Cole-McLaughlin K, Scorzelli G (2004) Multi-resolution computation and presentation of contour trees. In: Proceedings of the IASTED conference on Visualization, Imaging, and Image Processing, pp 452–290
Pascucci V, Scorzelli G, Bremer P-T, Mascarenhas A (2007) Robust on-line computation of Reeb graphs: simplicity and speed. ACM Trans Graph 26(3):58
Quadrat J-P, Lassere JB, Hiriart-Urruty J-B (2001) Pythagoras' theorem for areas. Amer Mathem Month 108(6):549–551
Reeb G (1946) Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique. Comptes Rendus Acad, Science Paris 222:847–849
Smale S (1961) On gradient dynamical systems. Ann Math 74:199–206
Takahashi S, Nielson GM Takeshima Y, Fujishiro I (2004a) Topological volume skeletonization using adaptive tetrahedralization. In: Proceedings of the Geometric Modeling and Processing 2004, pp 227–236
Takahashi S, Takeshima Y, Fujishiro I (2004b) Topological volume skeletonization and its application to transfer function design. Graph Models 66(1):24–49
Weber G, Dillard S, Carr H, Pascucci V, Hamann B (2007) Topology-controlled volume rendering. IEEE Trans Visual Comput Graphics 13(2):330–341
Zhou J, Takatsuka M (2009) Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. IEEE Trans Visual Comput Graphics 15(6):1481–1488