Volume Graphics (2006)
T. Möller, R. Machiraju, T. Ertl, M. Chen (Editors)

Contour Nest: A Two-dimensional Representation for Three-dimensional Isosurfaces

Shinobu Mizuta[1] Toshiki Ono[1] and Tetsuya Matsuda[1]

[1]Graduate School of Informatics, Kyoto University, Kyoto, JAPAN

Abstract

Visualization of the topological structure of isosurfaces such as exclusion and inclusion plays an important role in analyzing three-dimensional (3D) scalar data like medical images, especially in selecting isosurfaces to observe. Under certain conditions, all isosurfaces in a 3D scalar field are closed surfaces having nest structure. Our purpose here is to translate closed isosurfaces in 3D space as a nest structure.

Contour Tree (CT) can describe the nest structure among isosurfaces by tree structure, but it is difficult for users to understand the nest structure by direct observation of the tree shape. In this report, we propose a two-dimensional (2D) representation procedure named Contour Nest (CN) for intuitive display of the nest structure of isosurfaces. In the proposed method, nest structure of 3D isosurfaces extracted by Contour Tree is represented as 2D nest of rectangles. We have evaluated the effectiveness of the proposed method in selecting 3D isosurfaces.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Hierarchy and geometric transformations I.3.6 [Computer Graphics]: Interaction techniques

1. Introduction

With the advance of the imaging technology and the improvement of computer power, the opportunities of using three-dimensional (3D) scalar data are rapidly increasing. In the medical field, various types of 3D scalar data are used such as 3D X-ray computer tomography images, magnetic resonance images, and ultrasound images.

Visualization of the topological structure of isosurfaces such as exclusion and inclusion can play an important role in analyzing 3D scalar data, especially in selecting isosurfaces to observe. Under certain conditions, all isosurfaces in a 3D scalar field are closed surfaces having nest structure. Our purpose here is to translate closed isosurfaces in 3D space as a nest structure.

Contour Tree (CT) [BR63] [CSA03] and its variations are structuring procedures for isosurfaces of scalar fields. If isosurfaces of a 3D scalar field have a nest structure, CT can describe the structure by tree structure.

CTs can be used as interfaces to select isosurfaces for 3D volume visualization. Bajaj et al. use CTs in their system to decide field values [BPS97]. Carr et al. have constructed a system to select isosurfaces efficiently with interaction to CT [CS03]. In these cases, CTs are displayed as 2D trees. However, it is difficult for users to understand the nest structure by direct observation of the tree shape. In this report, we propose a two-dimensional (2D) representation procedure named Contour Nest (CN) for intuitive display of the nest structure of isosurfaces. In the proposed method, nest structure of 3D isosurfaces extracted by Contour Tree is represented as 2D nest of rectangles.

2. Contour Tree

Contour Tree (CT) is a tree-structured graph, representing the transitions of isosurfaces (appearance, disappearance, join and split) in continuous scalar fields, with increase or decrease of the threshold (isovalue) of field value [BR63]. Here, we define CT based on the references [CSA03] as follows:

- CT is a tree-structured graph having nodes and arcs.
- A node of CT represents a critical point (local maximum, local minimum or saddle) and the correspondent isosurface. Nodes and critical points satisfy one-to-one correspondence.

- An arc of CT links two nodes. The arc represents a region bounded by two isosurfaces corresponding to these two nodes. An arc and a region have a one-to-one relationship.

We call the nodes and arcs *supernodes* and *superarcs*, respectively. We also introduce additional nodes on the superarcs to represent isosurfaces in the regions corresponding to the superarcs. The isosurfaces do not include any critical points. We call these nodes *regular nodes*. *Nodes* consist of supernodes and regular nodes. We use the word *arcs* as the links between nodes. We call this type of CT Augumented Contour Tree (ACT) [CSA03]. CT can be extracted from scalar fields of any dimentions [CSA03].

In digital images, an isosurface for a threshold can be represented as a boundary between two regions of foreground and background, those are represented by connected pixels (voxels). Considering the characteristics of isosurfaces in digital images, we have proposed a modified ACT named Region-based Contour Tree (RBCT) [MM05]. In RBCT, an *arc* represents an isosurface, and a *node* represents a region bounded by isosurfaces. The region consists of a set of pixels or voxels having the same pixel (voxel) value in an image. Here we denote $U_i(T)$ as ith region corresponding to a node, having pixel (voxel) value T. A digital image has finite number of isosurfaces defined above, and RBCT can describe all the isosurfaces without redundancy.

Figure 1 illustrates an example of RBCT. Figure 1(a) represents a digital image, and Figure 1(b) is the RBCT corresponding to (a). Similarly as CT for continuous scalar fields, RBCT can be extracted from digital images of any dimension. In the rest of this report, CT is assumed to be RBCT.

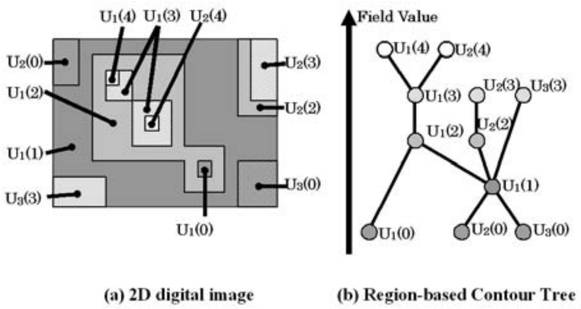


Figure 1: *Region-based Contour Tree.*

3. 2D display of Contour Trees using Contour Nests

Under certain conditions, all isosurfaces in a 3D scalar field are closed, having a nest structure. We propose a new method named *Contour Nest (CN)* to represent the nest structure, as the nest of 2D rectangles. CN is extracted from CT.

3.1. Introducing the isosurfaces surrounding whole images

If necessary, we can represent the isosurface that surrounds whole image, by setting the outside region of the image having pixel value ψ [MM05]. If ψ is smaller than the minimum pixel value of the image on outside boundary (e.g. $\psi = -\infty$) or larger than the maximum value (e.g. $\psi = +\infty$), the boundary can be treated as a closed isosurface that surrounds whole image.

In this condition, all other isosurfaces are closed, and the resulting CT become an equivalent tree structure as Inclusion Tree [MG00] . The resulting isosurfaces change in response to the value of ψ . Figure 2 is the CT extracted from Figure 1(a) with introduction of isosurface surrounding whole image, by setting $\psi = -\infty$. In this figure, *Virtual Node* represents the outside region of image. *Outside Arc* represents the isosurface surrounding the whole image. *Outside Node* represents a node corresponding to the pixels (voxels) on the outside boundary, having the minimum value.

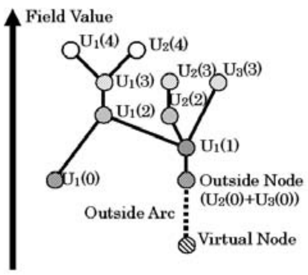


Figure 2: *Contour Tree for Figure 1(a) with an isosurface that surrounds whole image ($\psi = -\infty$).*

3.2. Construction of Contour Nests

Let $\gamma(e)$ be the region surrounded by an isosurface corresponding to the arc e on CT, and $S(e)$ be the size (area for 2D space and volume for 3D) of $\gamma(e)$. After introduction of the isosurface that surrounds whole image, CT can be considered as a rooted tree where Virtual Node is the root. Here, we can introduce the following theorems for all nodes excluding Virtual Node and the leaf nodes.

Theorem 1. *For the parent arc p and a child arc c of a node n,* $\gamma(p) \supset \gamma(c)$.

Theorem 2. *For the parent arc p and all child arcs* $c_i(i = 1,...,N, N \geq 1)$ *of p,* $S(p) > \sum_{i=1}^{N} S(c_i)$.

In the theorems above, we denote *parent arc p* of a node n as the arc connecting n and the parent node. Similarly, we denote *child arc c* of a node n as the arc connecting n and a child node. From these characteristics, CT can easily handle the nest structure of isosurfaces.

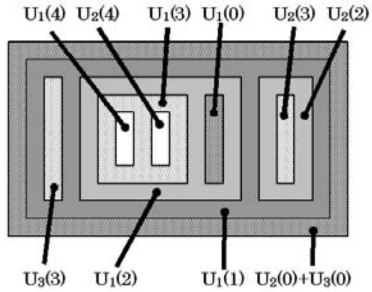


Figure 3: *Outline of Contour Nest for Figure 1(a) extracted from Contour Tree of Figure 2.*

If we illustrate a 2D rectangle $\delta(p)$ for the parent arc p of a node n where the area of the rectangle is proportional to $S(p)$, it is guaranteed that $\delta(c_i)$ for all child arcs $c_i(i = 1, ..., N, N \geq 1)$ of n can be illustrated in $\delta(n)$ without overlapping, from the theorems described above. Using this characteristic, we represent the nest structure of input data by transforming CT. Figure 3 illustrates the outline of CN viewing the nest structure of Figure 1(a), extracted from CT of Figure 2. In RBCT for digital images, $\gamma(e)$ of an arc e can be decided as the set of pixels (voxels) corresponding to the descendant nodes of e [MM05]. Therefore, $S(e)$ is easily calculated by counting the pixels (voxels) of $\gamma(e)$.

Various procedures have been proposed to represent tree structures in 2D [dOL03]. The proposed method is close to Treemaps [Shn92] and it can be considered as an application of Treemaps to CT.

3.3. Advantages of the proposed method

CN for 3D scalar data represents the nest structure of isosurfaces, as the nest structure of 2D rectangles. This procedure can be considered as the reduction of dimension from 3D to 2D, preserving the nest structure of isosurfaces. From this feature, CN is expected to be more intuitive in comparison with the direct representation of tree structure like CT.

4. Evaluation of the proposed method

4.1. Overview

In order to evaluate the effectiveness of Contour Nest, we have carried out experiments to measure the efficiency in selecting isosurfaces from 3D scalar data utilizing graphical user interface (GUI). The 3D data for experiments are simple 3D images, and the task is to select isosurfaces having topological characteristics.

4.2. Representation of the nest structure of isosurfaces

We have experimentally evaluated CT and CN as the representations of the nest structure of 3D isosurfaces. In the system for experiments, all nodes of CT are visible to users except for Virtual Node. The nodes can be considered to represent isosurfaces of the correspondent parent arcs.

Since the vertical positions of nodes in common representation of CT are decided from the correspondent field (voxel) values, crossing of arcs cannot always be avoided if cavities exist in the data [CS03]. However, if the positions follow the hierarchy of the CT, we can avoid the crossing.

In order to hierarchically re-layout the nodes of CT, we have introduced sizes (volumes) of regions surrounded by isosurfaces to decide the vertical positions of nodes. Besides the common CT and CN, we have evaluated this representation named Volume Contour Tree (VCT).

4.3. 3D Data for evaluation

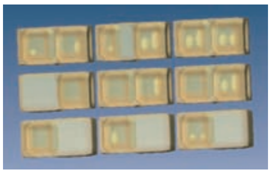
The dataset for the experiment consists of ten 3D digital images, which possibly include cavities. The isosurfaces of each image form nest structure having maximally 4 layers. All regions surrounded by the isosurfaces are rectangular parallelepiped. Table 1 is the condition of these data. Here, the layers are counted from outside to inside. In the 9 regions of Layer 2 of one image, the nest characteristics (numbers of regions) of Layer 3 and 4 are different from each other. The sizes of regions of Layer 3 and 4 are randomly decided under the condition of nest structure. The voxel values of these regions are also decided randomly. Figure 4 shows an image by 3D volume rendering and the nest structure of it represented by CT, VCT and CN.

Table 1: *Condition of data for evaluation.*

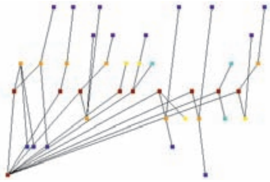
Number of regions in one region of outside layer	Layer 1	(1)
	Layer 2	9
	Layer 3	1 or 2
	Layer 4	0 or 1 or 2
Size of a region (voxels)	Layer 1	7525 (= $25 \times 43 \times 7$)
	Layer 2	455 (= $13 \times 7 \times 5$)
	Layer 3	15 or 45 or 75
	Layer 4	1 or 2 or 3
Voxel value	Layer 1	0
	Layer 2	3
	Layer 3	2 or 4
	Layer 4	0 or 1 or 5 or 6

4.4. Experimental condition

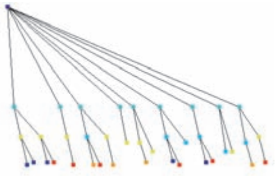
The experiments have been carried out for 5 subjects. The task is to select isosurfaces of Layer 2, satisfying the indicated conditions of the nest characteristics of surrounding isosurfaces. The conditions indicated to the subjects are the numbers of regions of Layer 3 and 4. The subjects have been asked to carry out the task, and the efficiency of the task has been evaluated from the time to select isosurfaces with the



(a)3D data



(b)Contour Tree



(c)Volume Contour Tree

(d)Contour Nest

Figure 4: *Example of 3D image and correspondent CT, VCT and CN.*

GUI systems utilizing CT, VCT and CN. In these representations, the horizontal positions of nodes or rectangles indicating the 9 regions of Layer 2 have been randomized.

All 10 images have been randomly shown once to user as a set of trial. Each subject has been asked to carry out the experiment for 11 sets, using the three systems. For each trial set, the three systems have been used in randomized order. The first and last set of 11 trials has not been used for the evaluations. Therefore, the total number of the trial evaluated is $10 \times 9 = 90$. The subjects have been asked to click button on the systems, at the start and the end of each trial. The time to select isosurfaces has been measured by sensing the two actions of clicking the button.

4.5. Experimental results

Table 2 shows the difference of the time to select isosurfaces in CT, VCT and CN. For all 5 subjects, the time using CN is shorter than VCT and CT. About the comparison between CN and CT, the P values are $P \leq 5.6 \times 10^{-7}\%$ for all subjects. About the comparison between CN and VCT, the P values are $P \leq 5.7\%$ excepting for subject E. These results indicate that selection of isosurfaces using CN is more efficient than those using other representations, especially CT.

5. Conclusion

The purpose of our research here is to translate closed isosurfaces in 3D space as the 2D nest structure, to select the isosurfaces. We have proposed a procedure named Contour Nest, which represents the nest structure of 3D isosurfces as the nest structure of 2D rectangles. From experimental comparison with the direct display of Contour Tree, we have confirmed the effectiveness of the proposed method.

Table 2: *Difference of the time to select isosurfaces in CT, VCT and CN.*

Subject	Procedure	Average process time(sec) (():standard deviation)	Difference to CN (P value)
A	CT	6.17(5.91)	9.80×10^{-9}
	VCT	3.57(1.11)	1.54×10^{-16}
	CN	2.31(0.69)	-
B	CT	8.55(5.44)	1.98×10^{-18}
	VCT	3.62(1.76)	5.10×10^{-10}
	CN	2.27(0.82)	-
C	CT	6.73(3.59)	4.59×10^{-18}
	VCT	3.93(1.79)	1.04×10^{-8}
	CN	2.65(1.00)	-
D	CT	6.95(3.56)	9.71×10^{-19}
	VCT	3.07(1.46)	5.70×10^{-2}
	CN	2.77(1.12)	-
E	CT	9.53(7.30)	5.54×10^{-9}
	VCT	4.76(2.31)	2.15×10^{-1}
	CN	4.46(2.76)	-

Acknowledgement A part of this research was supported by Grant-in-Aid for Scientific Research on Priority Areas (2) from Ministry of Education, Culture, Sports, Science and Technology, Japan (No. 15070206).

References

[BPS97] BAJAJ C. L., PASCUCCI V., SCHIKORE D. R.: The contour spectrum. *Proceedings of IEEE Visualization 1997* (1997), 167–173.

[BR63] BOYELL R., RUSTON I. H.: Hybrid techniques for real-time radar simulation. *IEEE Proceedings Fall Joint Computer Conference 63* (1963), 36–37.

[CS03] CARR H., SNOEYINK J.: Path seeds and flexible isosurfaces using topology for exploratory visualization. *Proceedings of IEEE TCVG symposium on Visualization* (2003), 49–59.

[CSA03] CARR H., SNOEYINK J., AXEN U.: Computing contour trees in all dimensions. *Computational Geometry 24* (2003), 73–94.

[dOL03] DE OLIVEIRA M. C. F., LEVKOWITZ H.: From visual data exploration to visual data mining: A survey. *IEEE Trans. on Visualization and Computer Graphics 9*, 3 (2003), 378–394.

[MG00] MONASSE P., GUICHARD F.: Scale-space from a level lines tree. *Journal of Visual Communication and Image Representation 11* (2000), 224–236.

[MM05] MIZUTA S., MATSUDA T.: Description of digital images by region-based contour trees. *Lecture Notes in Computer Science 3656* (2005), 549–558.

[Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. on Graphics 11*, 1 (1992), 92–99.