Interpolation-Based Extraction of Representative Isosurfaces

Oliver Fernandes, Steffen Frey and Thomas Ertl

University of Stuttgart
VISUS Visualization Research Institute
oliver.fernandes@visus.uni-stuttgart.de

Abstract. We propose a novel technique for the automatic, similarity-based selection of representative surfaces. While our technique can be applied to any set of manifolds, we particularly focus on isosurfaces from volume data. We select representatives from sets of surfaces stemming from varying isovalues or time-dependent data. For selection, our approach interpolates between surfaces using a minimum cost flow solver, and determines whether the interpolate adequately represents the actual surface in-between. For this, we employ the Hausdorff distance as an intuitive measure of the similarity of two components. In contrast to popular contour tree-based approaches which are limited to changes in topology, our approach also accounts for geometric deviations. For interactive visualization, we employ a combination of surface renderings and a graph view that depicts the selected surfaces and their relation. We finally demonstrate the applicability and utility of our approach by means of several data sets from different areas.

1 Introduction

The visual analysis of surfaces is an important task in many different domains, including a variety of medical applications and engineering. While the approach presented in this paper can deal with arbitrary surface representations, in the following, we mainly focus our discussion on isosurfaces generated from volume data. This data can be obtained through measurements via scanners (e.g. medical CTs, but also material testing for industrial applications), or simulations. Additionally, data may be static or time-dependent. While three dimensional scalar fields represent a common data type in scientific visualization, the complexity of these data sets increases steadily with their size. A default tool used to examine them is the generation of isosurfaces for a given threshold value. However, selecting threshold values showing the interesting features of the data set is aggravated by several problems. With the infinite possibilities of thresholds to choose from, manually identifying the more interesting isosurfaces can be very tedious.

In the following, we discuss our approach to determine characteristic isosurfaces based on transportation-based interpolation. We review related work in Sec. 2, and give on overview on our approach in Sec. 3. In particular, we contribute the following:

- our approach to determine interpolated surfaces and their similarity w.r.t. a reference (Sec. 4)
- the matching of similar surfaces across time or isolevels (Sec. 5)
- determining characteristic surfaces on the basis of the similarity between interpolate and reference (Sec. 6)

We evaluate our approach, and discuss its merits and limitations in Sec. 7. We finally conclude our work in Sec. 8.

2 Related Work

Isosurface Extraction and Rendering. For uniform grids based on trilinear interpolation, classical Marching Cubes (MC) [1] and variants are the most popular to explicitly extract isosurfaces, and are used as a basis for isosurfaces in this paper. Other approaches use Voronoi diagrams [2], advancing front techniques [3], and meshing from point clouds [4]. An overview on quad meshing techniques is given by Bommes et al. [5]. Theisel [6] represents the contours of a piecewise trilinear scalar field as trimmed surfaces of triangular rational cubic Bézier patches. For isosurface extraction from higher-order data, quad mesh generation techniques [7], contouring [8], and approximate isocontouring [9] have been proposed. Approaches for rendering implicit surfaces include BlobTrees [10] and raytracing with both interval and affine arithmetic [11].

Isosurface selection. A prominent approach for selecting characteristic representatives is the contour tree, which can be used to track the evolution of the topology of isosurfaces. A good overview of methods generating the graph is given by Biasotti et al. [12], and many improvements have been made towards efficiently employing the contour tree in arbitrary dimensions, e.g. Carr et al.[13]. Another approach is collecting statistical information on the scalar field [14][15], and selecting thresholds based on these results.

Isosurface similarity and morphing. Several methods for comparing surfaces for their similarity and consequentially morphing them have been proposed. With the Hausdorff distance being very intuitive and generally applicable, and in addition fast to compute [16], it is a choice similarity metric for two comparing two surfaces. The method in Bruckner et al. [17] enables automatic selection of isosurfaces based on an entropy similarity metric. It is noteworthy to mention, that while [17] generally takes a similar approach, our method differs as we do not necessarily require an underlying continuous representation as a scalar field. Unlike their entropy-based similarity metric (which requires an continuously defined/interpolated data set), our approach, based solely on the Hausdorff distance metric, works on arbitrary point cloud sets, including ones derived e.g. from analytically defined and higher order surfaces. The goal of the technique proposed by Wei et al. in [18] is to verify that a set of isosurfaces are sufficient to represent the entire scalar field. This is complementary to our technique, and may be used to set up the input set to our method.

3 Overview

In this work, we propose a novel technique for the automatic, similarity-based selection of representative surfaces, chosen from a set of surfaces constructed by varying a generating parameter like threshold value or time. Note that this 'base' set is acquired from a different source, e.g. using an isovalue threshold sweep, or employing complementary algorithms as mentioned in Sec. 2. We rely on two different similarity metrics in this work (Sec. 4): (1) the Hausdorff distance for fast computation, and (2) the scalable Minimum Cost Flow (MCF) Distance, which in addition to similarity also yields an surface interpolation scheme, used in the refinement step later on. First, our approach uses a low-accuracy, high-speed variant of the MCF Distance to do a comparison between connected components for consecutive thresholds. This establishes a set of so-called paths approximating the evolution of individual components similar to a contour tree (Sec. 5). In the refinement step, a component's change along a path is examined, by first calculating a linear interpolation between the first and the last component of a path using the MCF Interpolation. All components of the path are then compared to their appropriately evaluated interpolate using the fast and accurate Hausdorff distance. Provided the similarity distance exceeds a user-defined threshold, the path is subdivided at the deviating (and therefore representative) surface, and the sub-paths are retested.

4 Distances and Interpolation Between Isosurfaces

To decide if two arbitrary surfaces are similar, several metrics with different properties can be employed. In our approach we use two different schemes, Hausdorff distance and Minimum Cost Flow (MCF) Distance. We also employ a point cloud interpolation, which maps samples of a surface to samples of a different surface, invoking a MCF solver. In this section, we give a short overview of employed metrics and how the MCF Distance calculation yields an interpolation for two point clouds.

Hausdorff Distance. As mentioned earlier, we calculate the Hausdorff distance between surfaces, which is the supremum of the pairwise shortest distance from all points of one surface compared to the other. The mathematical definition also works on arbitrary point sets. One can easily see that taking a subset of points from the surfaces and calculating the Hausdorff distance for these will yield a good approximation for the surfaces themselves, while a uniform sampling with density based on largest surface area ensures a reasonable accuracy. Another important trait of the Hausdorff distance is that no further information is required, rendering it applicable for arbitrary geometry.

Minimum Cost Flow Distance. In this paragraph, we first explain the MCF problem, and then how it maps to a similarity function. Given two sets of nodes, sources and targets, weighted edges between sources and targets are established. A quantity Q is defined on all nodes, while the sources get positive values, the targets gets negative ones. The sum of Q over both the source and target set must

be exactly zero. The problem is now to move the quantity along the given edges, so that each node has zero quantity after the procedure. Depending on the edges available, this problem can be usually be solved in numerous ways. An additional condition can be imposed by requiring the so-called cost C to be minimal. C can be calculated by multiplying the weight of an edge with the amount of quantity moved across this edge, and summing this up for all edges participating in the solution. This severely reduces the number of possible solutions, very often the global minimum of C is unique.

The solution and it's cost C of a MCF problem can be mapped to a distance function between two point sets A and B using the following rules:

- Without limiting the generality of the mapping we declare the points in A to be the sources, and the points in B to be the targets.
- By connecting a point from A with a point from B we define an edge, and set its weight (i.e. cost) to the Euclidean distance between the points. This is done for all possible pairings.
- Each source node gets the quantity 1, each target receives -1.
- To fulfill the prerequisite of having a sum of exactly zero, the necessary amount of quantity (either positive or negative) gets distributed randomly to the set with less nodes (points).

Unlike the Hausdorff distance, the MCF distance additionally yields a direct point to point assignment, which is also useful for interpolation.

Minimum Cost Flow Interpolation: Executing the MCF algorithm will yield an assignment for each point of the set containing fewer points, to one or more points of the set containing more points. Per definition, all edges will be transporting exactly none or one unit (since both sets initially receive only 1 unit, either negative or positive). The assignments defined are simply the edges transporting a unit of the quantity. The Hausdorff distance definition may be used in a similar way, by assigning each point its closest neighbor of the other set. This will however introduce a heavy bias for certain points, e.g. the protruding peaks of a surface, since there is no limit on how many points are allowed to be mapped, which is avoided by the MCF solution.

5 Determining Component Evolution

As a preliminary similarity association, a coarse pre-matching is applied to connected components of consecutive isolevels, with the goal of associating a component $c_s \in C^s$ for an isolevel ρ_i to it's most similar candidate $c_t \in C^t$ for ρ_{i+1}. Performing this for all components at all isolevels will yield something similar to a contour tree, but based solely on point geometry (as opposed to topology). The matching of components for consecutive isolevels is done by executing the following two steps.

Determine Component Similarity: The first step is performed by comparing the connected components with each other. For each component $c_s, s \in 1 \dots n$ and $c_t, t \in 1 \dots m$, where n, m denotes the number of components in ρ_i, ρ_{i+1}

respectively, the surface is sampled uniformly (with respect to the surface area), but fairly coarse. All resulting point clouds of C^s are compared pairwise to all point clouds associated with C^t, and the MCF distance metric (as explained in subsection 4) is applied, with the calculated distance d resembling a (coarse) measure for their similarity.

Find Best Association: If the connected components are interpreted as nodes of a graph G, the pairs (c_s, c_t) can be interpreted as directed edges, with d being an associated edge value, and the direction being defined by increasing ρ. The first k edges, ordered by similarity d, are then added to the graph G, where $k = \max(n, m)$, while any edges containing a node already part of an edge in G are skipped. This leaves at the most one associating edge for each component in ρ_i and ρ_{i+1}. This is repeated for all isolevels $\rho_{\min} < \rho_i < \rho_{\max}$, and the graph G is defined containing all connected components of all isolevels as nodes, and edges connecting each node to their most similar component at the previous and next isolevel. The sub-graphs defined by a set of nodes which are connected by a series of edges will be called path P. Components which have exactly one edge, or none (i.e. the first and the last component c_f, c_l of a path), can be considered as candidates for representative surfaces. The resulting graph has some similarities to a contour tree, but additionally also has a few advantages. It already gives an impression of the similarity (determined as d) between components on a path P, which, in a contour tree, would simply be represented on a single edge. The above procedure will already yield representative surfaces similar to a contour tree. Even though this step will cover all correct matches for components, it might produce false positives. This can happen since the algorithm always picks a best match, even if there aren't any "true" matches left. In addition, a slight change of geometry on each increase of ρ_i can easily accumulate to a significant change of geometry from the first to the last node of a path. Hence a more accurate scheme is needed to augment this fast but coarse pre-selection.

6 Refining Selection of Characteristic Isosurfaces

Even though the information gathered in the first step described in section 5 already yields a significant set of representative surfaces (by choosing the first and last components of established paths), potentially interesting candidates could be missed within a path, and false connections might still be in there. Since the first and last nodes of a path P are already marked as representative components, the intermediate nodes now need to be examined. To find other potential candidates $c_o \in P$, which differ significantly from both the first or last component, a linear interpolation scheme is executed and Hausdorff similarity metric applied. The following scheme is iterated on each path P to find further candidates.

Determine Similarity by Interpolation: More specifically, to determine if an original surface c_o shows enough similarity to both the first and last component c_f and c_l in a path P, a linear interpolation is performed between point clouds derived from c_f and c_l as explained in 4. The employed point clouds are again uniformly distributed samples, but unlike in Sec. 5, the resampling here is fairly

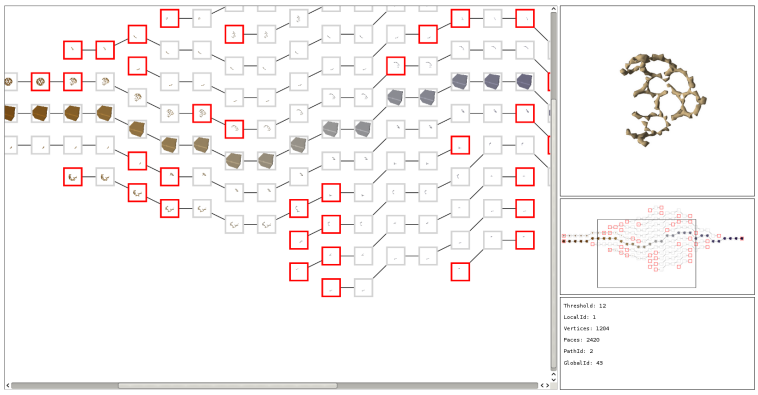
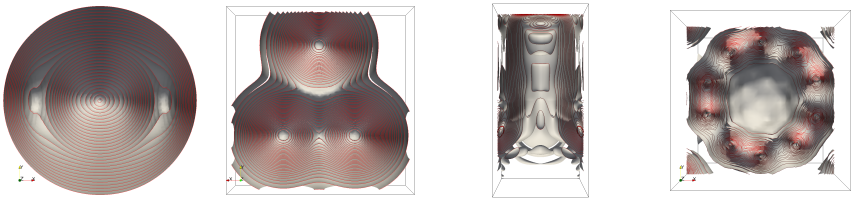


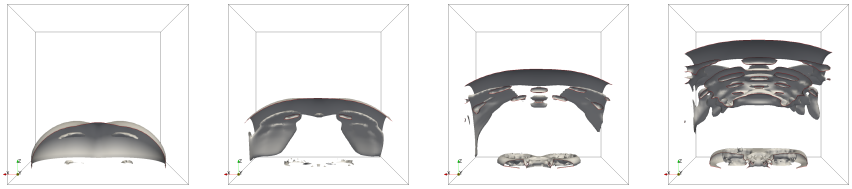
Fig. 1: Screen capture of the interactive graph tool. The main view shows the graph in detail with the nodes showing renders of the appropriate component. The top right view shows a detailed render of a selected node, while the lower shows statistical of a node or path. The overview window to the middle right assists in navigating the graph.

dense (with respect to surface areas of c_f and c_l), to ensure a reasonably accurate interpolation between the point clouds of the components.
The linear interpolation c_i is then evaluated at an interpolation parameter t based on the square root a of the surface areas of c_f, c_l and c_o using $t = (a_o - a_f)/(a_l - a_f)$. This ensures that, if a surface changes exactly linear, it will also be perfectly interpolated by c_i.
Conversely, the Hausdorff distance d_{oi} between original component c_o and interpolated component c_i is a measure for geometrical deviations from c_l, c_f, which might change the visual appearance of the component significantly.

Subdivide Paths: If any of the comparisons d_{oi} for the interpolated component c_i to the corresponding original component c_o yields a difference greater than a user-defined threshold ϵ, the deviation is considered significant, and the path needs to be subdivided. To reach a meaningful subdivision, the tested sub-path is increased incrementally. This means, a sub-path starting at the first $c_{f'} = c_f$ and ending two nodes along the path, at $c_{l'} = c_{f+2}$, is defined. If all nodes c_i between $c_{f'}$ and $c_{l'}$ fulfill the interpolation similarity as defined above, a node is added to the sub-path, $l' = l' + 1$, and the entire sub-path is retested. If the test fails at any given intermediate node c_i, the path is subdivided at the current l'. The procedure is reiterated and sub-path now starts on the first unsuccessfully added node, setting $c_{f'} = c_{l'}$, and $c_{l'} = c_{l'+2}$. The algorithm completes when $l' \geq l$, i.e. the current sub-path's end surpasses path P's end. The nodes $c_{f'}, c_{l'}$ of each sub-path are added to the characteristic set S. Note that per construction, all nodes on the sub-paths can be approximated by linear interpolation from the characteristic nodes (within the error of ϵ).



(a) Distorted Sphere (b) Gauss Blob (c) Rayleigh-Taylor (d) Bucky Ball

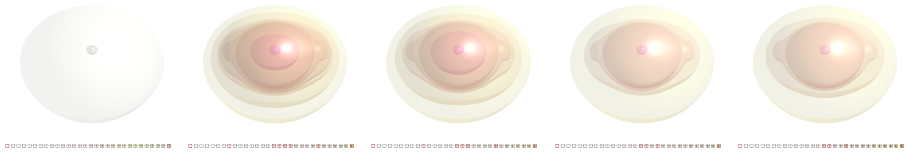


(e) 5jets ts100 (f) 5jets ts160 (g) 5jets ts230 (h) 5jets ts300

Fig. 2: Images giving an impression for the input data sets. a) A radially increasing scalar field, with an added distortion in x direction. b) Three gauss functions of varying intensity summed up to produce the scalar field. c) Time step 14 of the Rayleigh-Taylor data set. d) The C60 molecule in a scalar data field representation. e)-h) A fixed threshold generates surfaces for various time steps of the 5jets data set.

7 Results

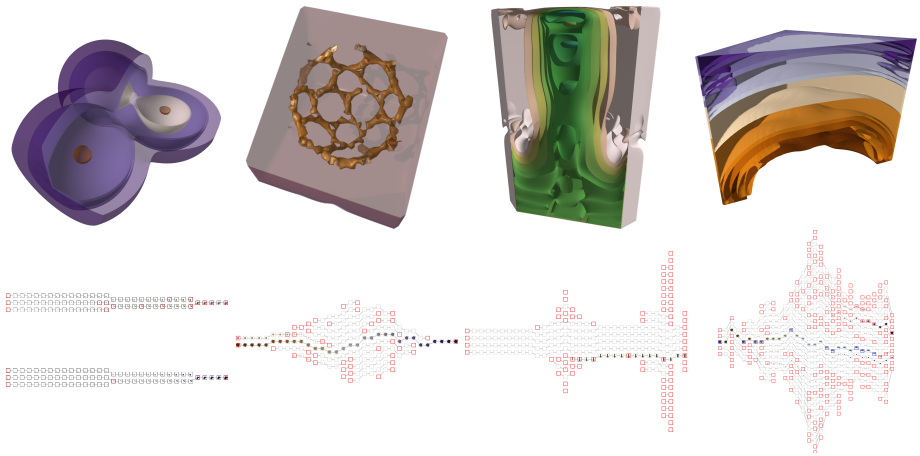
To demonstrate the usability of the approach, we applied the technique to several data sets. Data set size range from 64^3 to $128 \times 128 \times 256$. An impression of the input data sets is given in Fig. 2. The renderings are clipped to better see the contours for a subset of the isosurfaces. In all data sets for which the threshold value is varied, it ranges from minimum to maximum scalar value of the corresponding data set (except for *Bucky Ball*, see below), on $30 - 32$ intervals. For the *5jets* data set, every tenth time step was used from time step 100 to 300. All calculations were performed on a Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz.For each data set, we give a render of the complete set of selected isosurface components (Fig. 4), clipped to better show the results. Below the renderings of all data sets the graph G is drawn, showing components as square nodes, sorted by isovalue from left to right, and longer paths closer to the center on the vertical axis. Components selected by the algorithm are highlighted in red. Edges show the preliminary paths established.The graph is also interactive, and may be used to acquire additional information about components and paths, as seen in Fig. 1. For this, paths can be selected, and while selected nodes are highlighted with a blue outline, associated data is displayed on the right. A detail view of the component is shown for the selected node. The graph visualization implemented is only a simple tool to verify the most important results (see Future



(a) Cont. tree (b) $\epsilon = 0.5$ (c) $\epsilon = 1.0$ (d) $\epsilon = 2.0$ (e) $\epsilon = 4.0$

Fig. 3: Our algorithm applied to the *Distorted Sphere* data set for varying distance thresholds ϵ, in units of cell size, with a) showing the contour tree result for comparison. As seen for the second inner sphere in b), a small ϵ reacts earlier to the change from spherical to elliptical compared to c). As ϵ increases, less surfaces get selected. The salient surface in the middle range always gets selected.

Work, Section 8), and can easily be improved to query data from the input set.
Distorted Sphere: This data set, shown in Fig. 3, is the first synthetic data set, and serves to give an impression of how the user-defined Hausdorff distance error ϵ affects the selection of isosurfaces. As is to be expected, fewer isolevels are selected for increasing ϵ. Obviously the technique selects more isosurfaces characteristic to the data set than the contour tree, which would simply be two nodes for maximum/minimum isovalue, missing the salient surface in between.
Gauss Blob:Fig. 4a shows the second synthetic data set. This data set highlights how the algorithm handles changes in the contour tree. As can be easily be discerned from the graph, the first step creates the "contour tree", encoded in the node connections. Even though the algorithm does not show the actual merging (like a contour tree would), it successfully determines all involved components as characteristic, as well as selecting a few additional isovalues, since they differ enough from the surfaces associated with topology changes.
Rayleigh-Taylor: To get a clear view of the results obtained for the Rayleigh-Taylor instability, the surfaces have been rendered opaque, and a clipping plane was introduced. Comparing Fig. 4c) with the input (see Fig. 2c)) one can see that many cluttering surfaces have been removed. However, the most distinct features are still visible, as well a few supporting isolevels selected by our algorithm. The corresponding graph can be used to further investigate the selected surfaces.
5jets: Being a time series of isosurfaces, intersecting surfaces may occur, which however get handled by the algorithm directly. Since the myriad of surfaces would severely hinder exploration due to occlusion, we have picked a component (i.e. an edge in the contour tree) in the interactive graph and show its evolution over several isolevels (Fig. 4d), in terms of the surfaces selected by our algorithm. The corresponding path is shown on the graph below.
Bucky Ball: For this dataset, a sub-range of thresholds was chosen as input, where the main feature of the data set disintegrates into smaller components. As can be seen in Fig. 4b, the boundary regions form a path dominating the graph. The splitting of the main feature components, as well as the evolution of the sub



(a) Gauss Blob (b) Bucky Ball (c) Rayleigh-Taylor (d) 5jets

Fig. 4: Isosurfaces selected by our algorithm for the respective data sets, including the generated graph. For the time series data in 4d), a contour tree edge is chosen (blue path in graph) and the isosurfaces selected by the scheme are shown. The Gauss example 4a) includes a contour tree (lower graph) for comparison.

components can be easily extracted from the graph. Note that most selections are topology changes, correctly identified as characteristic surfaces.

8 Conclusion

We proposed a novel technique for automatically selecting a set of representative surfaces according to a minimum cost flow-based similarity metric. While our approach works for arbitrary sets of surfaces, we focussed on isosurfaces in the context of this paper. Here, we changed either the threshold value for a fixed time, or the time was varied for a fixed threshold. We demonstrated that our technique enabled a detailed selection of representative isosurfaces, based on the changes in geometry, as the isosurface threshold is varied. To achieve this, connected components of isosurfaces with increasing threshold are matched using a similarity measure derived from the cost of matching points of the surface with a minimum cost flow algorithm. However, even though geometrical changes accumulate over several steps, the individual distances cannot be simply added.We remedy this by interpolating a component's surface over a threshold range, employing the minimum cost flow algorithm again to obtain the interpolation. Based on the Hausdorff distance of the interpolated surface to the original, additional threshold values are added to the representative set.
For future work, the currently employed simple sampling strategy can be easily improved, to guarantee a good approximation of the surface by the point cloud.

The selection scheme can be directly improved by entering other factors into the similarity calculation besides Hausdorff distance, e.g. employing change of curvature. Supplementing the graph tool with a query-based filtering of paths and components selected by the algorithm, would further enhance the utility as an interactive interface for exploration.

Acknowledgements: This work was primarily funded by Deutsche Forschungsgemeinschaft (DFG) under grant SPP 1648 (ExaScaleFSA).

References

1. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. Computer Graphics **21** (1987) 163–169
2. Dey, T., Levine, J.: Delaunay meshing of isosurfaces. In: Shape Modeling and Applications, 2007. (2007) 241 –250
3. Schreiner, J., Scheidegger, C., Silva, C.: High-quality extraction of isosurfaces from regular and irregular grids. TVCG **12** (2006) 1205–1212
4. Scheidegger, C.E., Fleishman, S., Silva, C.T.: Triangulating point set surfaces with bounded error. In: EG symposium on Geometry processing. (2005)
5. Bommes, D., Lévy, B., Pietroni, N., Puppo, E., Silva, C., Tarini, M., Zorin, D.: Quad meshing. In: Eurographics, The Eurographics Association (2012) 159–182
6. Theisel, H.: Exact isosurfaces for marching cubes. Computer Graphics Forum **21** (2002) 19–32
7. Remacle, J.F., Henrotte, F., Baudouin, T., Geuzaine, C., Bchet, E., Mouton, T., Marchandise, E.: A frontal delaunay quad mesh generator. In: 20th Meshing Roundtable. (2012) 455–472
8. Wiley, D.F., Childs, H.R., Gregorski, B.F., Hamann, B., Joy, K.I.: Contouring curved quadratic elements. In: VisSym. (2003) –1–1
9. Pagot, C.A., Vollrath, J., Sadlo, F., Weiskopf, D., Ertl, T., Comba, J.: Interactive isocontouring of high-order surfaces. In: Scientific Visualization. (2011)
10. Shirazian, P., Wyvill, B., Duprat, J.L.: Polygonization of implicit surfaces on multi-core architectures with simd instructions. In: EGPGV. (2012) 89–98
11. Knoll, A., Hijazi, Y., Kensler, A., Schott, M., Hansen, C.D., Hagen, H.: Fast ray tracing of arbitrary implicit surfaces. CGF **28** (2009) 26–40
12. Biasotti, S., De Floriani, L., Falcidieno, B., Frosini, P., Giorgi, D., Landi, C., Papaleo, L., Spagnuolo, M.: Describing shapes by geometrical-topological properties of real functions. ACM Comput. Surv. **40** (2008) 12:1–12:87
13. Carr, H., Snoeyink, J., van de Panne, M.: Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. CGTA **43** (2010) 42–58
14. Khoury, M., Wenger, R.: On the fractal dimension of isosurfaces. IEEE Transactions on Visualization and Computer Graphics **16** (2010) 1198–1205
15. Tenginakai, S., Lee, J., Machiraju, R.: Salient iso-surface detection with model-independent statistical signatures. In: IEEE Visualization. (2001)
16. Tang, M., Lee, M., Kim, Y.J.: Interactive hausdorff distance computation for general polygonal models. ACM Trans. Graph. **28** (2009) 74:1–74:9
17. Bruckner, S., Möller, T.: Isosurface similarity maps. Computer Graphics Forum **29** (2010) 773–782 EuroVis 2010 Best Paper Award.
18. Wei, T.H., Lee, T.Y., Shen, H.W.: Evaluating isosurfaces with level-set-based information maps. Computer Graphics Forum **32** (2013) 1–10