

Identifying Nearly Equally Spaced Isosurfaces for Volumetric Data Sets

Martin Imre^a, Jun Tao^a, Chaoli Wang^a

^aDepartment of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556

Abstract

Isosurfaces are an important visual representation of volumetric data sets and isosurface extraction and rendering remains one of the most popular methods for volume visualization. Previous works identify a small set of representative isosurfaces from a set of sample ones, providing a concise description of the underlying volume. However, these methods do not lend themselves to equally spaced isosurfaces, i.e., keeping the same distance between neighboring isosurfaces, which can be advantageous from the user's perspective in terms of visual summarization and interactive exploration. In this paper, we present a new solution that efficiently identifies a set of nearly equally spaced isosurfaces for a given volume data set. Our approach includes an estimation stage of linear interpolation and a refinement stage of binary search in order to balance the tradeoff between quality and performance. The refinement stage can incorporate spike and/or jump treatments to possibly improve the convergence. Experimenting with multiple data sets of different sizes and characteristics, we perform both quantitative and qualitative studies, demonstrate the efficiency and effectiveness of our approach, and summarize our findings.

Keywords: Volumetric data, Isosurfaces, Topology, Equally spaced, Similarity map

1. Introduction

Numerical simulations are extensively used by scientists to observe various phenomena that are not easily captured by real experiments. These simulations normally produce an ample amount of data, requiring effective tools to visualize and analyze them. A typical visualization presents the simulation results as a series of volumes. One of the essential techniques to gain insights into these volumes is isosurface rendering. To describe the structure of a volume, one can extract and visualize isosurfaces. These surfaces describe surface geometries with all points sharing the same isovalue. For insightful visualization, it is critical to select a set of salient isosurfaces that captures different features and characteristics of the underlying volume.

One common solution is to select a set of distinctive or representative isosurfaces from sample ones based on a certain similarity measure. For example, Tenginakai et al. [1] measured the similarities between isosurfaces using data histograms and higher order moments. Bruckner and Möller [2] derived distance fields from the sample isosurfaces and utilized mutual information to evaluate the similarity between the distance fields. The similarity values are organized in a matrix form named *isosurface similarity map* from which the representative isosurfaces are selected.

One major challenge exists for these approaches: it is essential for them to start with a set of reasonably good sample isosurfaces that capture different features in a balanced way. Otherwise, the features missing in the samples will not be recovered in the later stages, or the selection may be biased by favoring the features corresponding to more samples. However, straightforward sampling techniques do not guarantee the desired set of samples. Uniform sampling is likely to miss some

features when many of them reside in a small value range. Although sampling according to histograms of voxel values, i.e., placing more samples in the value ranges with more voxels, may alleviate this problem to some degree, it still suffers from oversampling as the value ranges with more voxels do not necessarily indicate more distinctive features.

Another key challenge is posed by the scale and complexity of the data generated by numerical simulations. To obtain a comprehensive understanding of physical phenomena, the simulations usually involve multiple variables and their interactions over time, resulting in large-scale time-varying multivariate volume data sets. This requires a surface-based analysis to be efficient in two aspects. First, the distance between two isosurfaces should be measured efficiently. Second, the number of distance calculations should be minimized so that one can afford to take a full run and draw a complete picture of the data. Previous approaches focused more on the definitions of similarity measures and less on performance optimization. For example, it took around 25 minutes to process a single volume with the isosurface similarity map approach [2]. This cost becomes prohibitive when analyzing a typical time-varying multivariate data set with tens of variables and hundreds of time steps.

In this paper, we present an approach for identifying nearly equally spaced isosurfaces, so that the distance between neighboring surfaces is as similar as possible to the average distance. In flow visualization, creating evenly spaced or mutually distant streamlines or stream surfaces has been well studied [3, 4, 5]. However, to the best of our knowledge, creating equally or evenly spaced isosurfaces has not been investigated. Our solution ensures that the isosurfaces corresponding to neighboring isovalues are distinct enough according to the given distance measure. When identifying a small number of isosurfaces, we

64 can consider the resulting isosurfaces as salient features on their
65 own. Compared to the similarity-based approaches for identi-
66 fying representative isosurfaces, our approach does not require
67 the isosurfaces to be selected from a limited set of sample ones.
68 It not only has a wider search space but also explicitly controls
69 over the resulting isosurfaces, which can potentially lead to bet-
70 ter results. Compared to the topology-based approaches, our
71 method is more flexible when equipped with different distance
72 measures. Although this offers great flexibility, our method re-
73 lies on features being a function of isovalues. Given that pre-
74 condition, our algorithm may capture the topological changes if
75 the distance measure is topology-aware, and it may produce iso-
76 surfaces with distinct shapes if the distance measure is shape-
77 aware. In addition, when a large set of isosurfaces is identified,
78 the results can serve as reliable input to other volume analysis
79 and visualization tasks. In our experiment, we find that tak-
80 ing our results as the input, the representative isosurfaces se-
81 lected using isosurface similarity map [2] and k-means [6] can
82 be improved. The comparison results will be presented in Sec-
83 tion 4.2.

84 Our approach includes two stages: an estimation stage that
85 quickly converges to a rough solution within a few iterations,
86 and a refinement stage that optimizes the estimation. For both
87 stages, only the distances between neighboring isosurfaces are
88 needed at each iteration. Leveraging the parallel computation
89 of GPU, we can process each iteration efficiently. In addition,
90 our approach can be flexibly customized with various distance
91 measures to meet different needs. In our experiment, we com-
92 pare the performance and sampling results using the isosurface
93 similarity map (ISM) measure [2] and the mean of the closest
94 point distances (MCP) [7].

95 The contributions of our work are as follows.

- 96 • First, we present a feasible solution to identifying nearly
97 equally spaced isosurfaces, an important yet seldom in-
98 vestigated problem. We shall see that our solution does
99 not fully converge in general but we are able to find a
100 solution with acceptable quality and performance trade-
101 off. Compared to similarity- or topology-based methods,
102 the set of isosurfaces generated by our method provides
103 an advantageous visual summarization of the volumetric
104 data, especially when the number of surfaces is small.
- 105 • Second, we perform a thorough study to compare param-
106 eter choices, distance measures, and qualitative results,
107 followed by a list of findings for other researchers to fol-
108 low. The proposed solution can be adopted by others for
109 incorporation into their high-performance volume data
110 analysis and visualization workflow.

111 2. Related Work

112 To analyze and visualize volumetric data sets, researchers
113 have sought different kinds of methods to understand the struc-
114 tures of volumes. The *distribution-based* methods focus on the
115 distributions of certain properties of the volume and identify the

116 salient structure based on their corresponding statistical char-
117 acteristics. The *topology-based* methods analyze the topologi-
118 cal structure of the volumes and highlight the structures corre-
119 sponding to topological changes. The *similarity-based* methods
120 measure the similarity between volume representations such as
121 isosurfaces and derive the representative ones based on their
122 similarities.

123 **Distribution-based methods.** Understanding the relation-
124 ships between the volume distribution and the isosurfaces al-
125 lows us to identify salient features. For instance, Tenginakai et
126 al. [1] detected salient isosurfaces using local higher order mo-
127 ments (LHOMs). LHOMs are computed and plotted for differ-
128 ent sample values for a semi-automatic selection. Scheidegger
129 et al. [8] applied Federer’s Coarea Formula to improve the iso-
130 surface statistics by weighting with the inverse gradient mag-
131 nitude. Duffy et al. [9] developed a mathematical model for
132 continuous functions and proved the convergence to continu-
133 ous statistics for regular lattices. Pekar et al. [10] proposed to
134 use Laplacian weighted histograms for significant isovalue de-
135 tection. However, the distribution of a volume data set does not
136 translate to the spatial relationship among surfaces extracted,
137 which is the focus of this paper.

138 **Topology-based methods.** These methods extract struc-
139 tures that essentially characterize properties of space such as
140 convergence, connectedness, and continuity, providing a con-
141 cise description of the overall structure of a volume. Bajaj
142 et al. [11] proposed the contour spectrum, an interface com-
143 bining the contour tree together with a variety of isosurface
144 statistics, such as area and enclosed volume. Bremer et al.
145 [12] presented the cancellation tree for describing the simpli-
146 fication of a Morse-Smale complex. Each simplification step
147 cancels a pair of critical points, i.e., minima and maxima. The
148 cancellation tree encodes the simplification steps and provides
149 the connections among critical points. They further extended
150 this approach to the hierarchical merge tree, which is a track-
151 ing graph that describes the temporal evolution of features [13].
152 Carr et al. [14] proposed to use the contour tree to encode the
153 nesting relationships among isosurfaces. It also serves as an in-
154 terface that allows users to select contours for operations such
155 as removal, evolution, and tracking. Correa et al. [15] intro-
156 duced the topological spine that connects critical points along
157 the steepest ascending or descending directions. In addition, it
158 includes geometric and contour nesting information, providing
159 better spatial reasoning.

160 Although rigorous, topology-based methods normally cap-
161 ture minute topological changes, which lead to a large number
162 of isosurfaces for a volume with complex topological variations.
163 This, however, may not always be necessary for users to un-
164 derstand the overall structure of the volume. In contrast, our
165 approach generates a small set of nearly equally spaced iso-
166 surfaces which are more amenable for user observation: each
167 surface is distinct enough and they are mutually distant in the
168 space. Such a set of isosurfaces could also be useful as a visual
169 summarization of the underlying volume.

170 **Similarity-based methods.** Recent works often seek to
171 measure the similarities between a set of sample isosurfaces and
172 derive the structure of the entire volume. For example, Bruck-

ner and Möller [2] evaluated the similarity between isosurfaces and organized them in the form of an isosurface similarity map. The similarity between two isosurfaces is defined as the mutual information shared by the distance fields of the two isosurfaces. Representative isosurfaces are identified using the isosurface similarity map, which stores all pairwise similarity values. Haidacher et al. [16] extended this approach to compare isosurfaces extracted from multiple volumes. Wei et al. [17] proposed a similarity measure between two isosurfaces based on intermediate level-set surfaces. The values on the intermediate surfaces are sampled from the volume and their entropy values are used to evaluate whether the level-set surfaces align well with the intermediate isosurfaces. Recently, Ma et al. [6] used a tensor-based perceptual distance measure that simulates the human visual system and employed k-means clustering to select representative isosurfaces for comparing different volumetric data compression approaches.

These methods, including our approach, require distance fields of isosurfaces to be computed. As a common need of many applications, accelerating the distance field computation has been extensively studied [18, 19, 20, 21, 22]. Yu et al. [22] presented the parallel distance tree that distributes the workload to multiple processors guided by a coarse global distance tree. Each processor then constructs a local distance tree and derives the distance fields. To compute the distance field, the bounding volume hierarchy (BVH)-tree is often used to identify the closest points. Liu and Kim [20] proposed the multi-BVH that combines the octree and BVH-tree. The use of octree provides additional information to reduce the number of BVHs to be traversed. Karras [23] introduced a GPU-based method to construct BVH-trees in parallel, which is by far the fastest GPU solution available. In Section 3.4, we will discuss how we use Karras’ algorithm to accelerate distance computation.

3. Our Approach

We propose a two-stage approach for finding nearly equally spaced isosurfaces. Both stages run over several iterations aiming for convergence. First of all, during the *estimation stage*, we measure the distance between every pair of neighboring isosurfaces and resample the isovalues based on these distances using linear interpolation. This stage, however, assumes piecewise linearity of the distance function between neighboring isosurfaces, which does not hold in general. In our experiment, it approaches approximate solutions in a few iterations but normally does not converge to the optimal solution. Therefore, we introduce the *refinement stage* that adopts a binary search strategy to adjust each isovalue so that its surface has equal distance to its two neighbors. By repeating this process for several iterations, we achieve nearly equal distances between all neighboring isosurfaces.

In this section, we denote a set of isovalues at iteration τ as $V^\tau = \{v_1^\tau, v_2^\tau, \dots, v_n^\tau\}$, the isosurface corresponding to an isovalue v_i^τ as s_i^τ , and the distance between two values v_i^τ and v_j^τ as the distance between their respective surfaces, $d_{i,j}^\tau$, or more generally, $d(v_i, v_j)$.

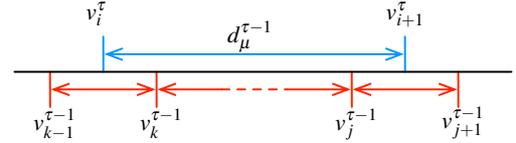


Figure 1: The estimation stage. Identifying isovalue v_{i+1}^τ based on its left neighbor v_i^τ and the distances evaluated at the previous time step $\tau - 1$.

3.1. Estimation Stage

For the estimation stage, we start from a uniformly sampled set of isovalues V^0 and gradually adapt the isovalues based on the previous set of isovalues and their distances. Specifically, at each iteration τ , we first approximate the isosurface using the approximation scheme of Imre et al. [24] (refer to Section 3.4), $s_i^{\tau-1}$ of each isovalue $v_i^{\tau-1}$ at the previous iteration $\tau - 1$ and evaluate the distance $d_{i,i+1}^{\tau-1}$ between every pair of neighboring isovalues $v_i^{\tau-1}$ and $v_{i+1}^{\tau-1}$ using their corresponding approximated isosurfaces. The average distance $d_\mu^{\tau-1} = (\sum_{i=1}^{n-1} d_{i,i+1}^{\tau-1}) / (n - 1)$ (μ stands for the average) is considered to be the target distance to achieve at the current iteration τ .

Then, starting from the first isovalue $v_i^\tau = v_{\min}$, where v_{\min} is the minimum isovalue, we resample each isovalue v_{i+1}^τ that has approximately the target distance $d_\mu^{\tau-1}$ to its left neighbor v_i^τ , as illustrated in Figure 1. The distance is estimated under two assumptions. First, the distance between neighboring isovalues can be linearly interpolated. For example, the isovalue v_i^τ falls between two previous isovalues $v_{k-1}^{\tau-1}$ and $v_k^{\tau-1}$. We assume that the distance between v_i^τ and $v_k^{\tau-1}$ can be linearly interpolated using the distance $d_{k-1,k}^{\tau-1}$. Second, we assume that the distance can be added to estimate the distance between non-neighboring isovalues. For example, we assume that the distance $d_{k,j}^{\tau-1}$ can be obtained using the summation of all neighboring distances between $v_k^{\tau-1}$ and $v_j^{\tau-1}$, i.e., $d_{k,j}^{\tau-1} = d_{k,k+1}^{\tau-1} + \dots + d_{j-1,j}^{\tau-1}$. In this way, we can iteratively identify the entire set of isovalues V_τ . This process can also be considered as a parameterization based on the distances evaluated from previous neighboring isovalues.

We repeat this process for several iterations until a predefined minimum iteration number δ_e is reached and the variation of neighboring distances stops decreasing. As shown in the first two rows of Figure 5, we can see that the estimation stage approaches the desired solution within a small number of iterations. Note that the computation of distances between neighbors, which is the most costly step, can be performed in parallel for each iteration. As previously mentioned, this stage is unlikely to converge since the two aforementioned assumptions do not hold for many volumetric data sets. In most cases, it is more likely to have $d_{i,j} + d_{j,k} > d_{i,k}$ due to the triangle inequality. Therefore, the estimation stage only provides a rough solution, and an additional refinement stage is needed to obtain the optimal solution.

3.2. Refinement Stage

In the refinement stage, we advocate a binary search strategy: placing the candidate isovalue in the middle of its two neighbors to identify an isosurface having the equal distance to its two neighboring isosurfaces. The distance function in this

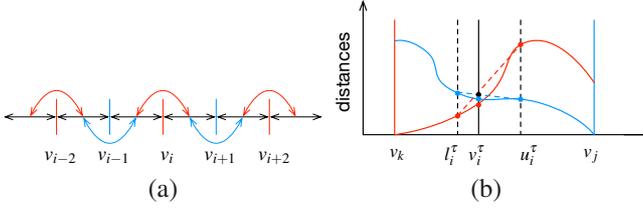


Figure 2: The refinement stage. (a) Adjusting odd-indexed and even-indexed isovalues alternatively. (b) Identifying an isovalue v_i that has the equal distance to its two neighbors v_k and v_j . The red (blue) curve in (b) represents the distance function from an isovalue in $[v_k, v_j]$ to v_k (v_j).

274 stage neither assumes linearity nor violates the triangle inequality. Unlike the estimation stage, this stage provides a slower
 275 but more robust process of convergence. This is achieved by
 276 adjusting the odd-indexed isovalues and even-indexed ones al-
 277 ternatively, as shown in Figure 2 (a). Specifically, the refine-
 278 ment stage is performed in multiple steps (δ_r). At odd steps, we
 279 adjust the red isovalues v_{i-2}, v_i, v_{i+2} with odd indices (assum-
 280 ing i is an odd number), so that they have the equal distance to
 281 their neighbor isovalues, i.e., $d_{i-3, i-2} = d_{i-2, i-1}$, $d_{i-1, i} = d_{i, i+1}$,
 282 and $d_{i+1, i+2} = d_{i+2, i+3}$, as indicated by the red arrows. Since
 283 the blue isovalues with even indices are fixed at odd steps, each
 284 odd-indexed isovalue can be adjusted independently in parallel.
 285 At even steps, we adjust the blue isovalues in the same fashion.
 286 Note that the blue and red arrows connect all distances between
 287 neighboring isovalues, which leads to equally spaced isovalues
 288 when this stage converges.

290 In every step, we use several iterations (δ_τ) of a binary
 291 search strategy to identify an isovalue v_i that has the equal dis-
 292 tance to its two neighbors v_{i-1} and v_{i+1} , as illustrated in Fig-
 293 ure 2 (b). This means that the goal becomes finding one inter-
 294 section point of the red and blue curves. At each iteration τ ,
 295 we maintain a lower bound l_i^τ and an upper bound u_i^τ that con-
 296 tain the intersection point between them. The lower and upper
 297 bound are initialized as v_k and v_j , respectively, i.e., $l_i^0 = v_k$ and
 298 $u_i^0 = v_j$. The lower bound maintains a property that it is always
 299 closer to v_k than v_j , i.e., $d_{k, l_i} < d_{l_i, j}$, and the upper bound main-
 300 tains a similar property in the opposite way, i.e., $d_{k, u_i} > d_{u_i, j}$.
 301 Due to these properties, the red and blue curves must intersect
 302 somewhere in the middle as long as the distance functions are
 303 continuous.

304 At each iteration, we assume that the two distance functions
 305 change linearly between the bounds, as shown by the blue and
 306 red dashed lines in Figure 2 (b), and compute the intersection
 307 point, as indicated by the black dot. This intersection point
 308 provides the new isovalue v_i^τ at step τ , as shown by the black
 309 solid line in Figure 2 (b). We compute the distances $d_{k, i}^\tau$ and
 310 $d_{i, j}^\tau$ and determine whether v_i^τ will replace the lower or upper
 311 bound. In this example, since $d_{k, i}^\tau < d_{i, j}^\tau$, indicated by the inter-
 312 section points between the black solid line and the two curves,
 313 we replace the lower bound with v_i^τ , i.e., $l_i^{\tau+1} = v_i^\tau$, so that the
 314 properties of the lower and upper bounds still hold.

315 It is clear that the smaller the search range gets, the better
 316 the distance functions can be approximated by linear functions.
 317 As shown in Figure 2 (b), the curves between the bounds are

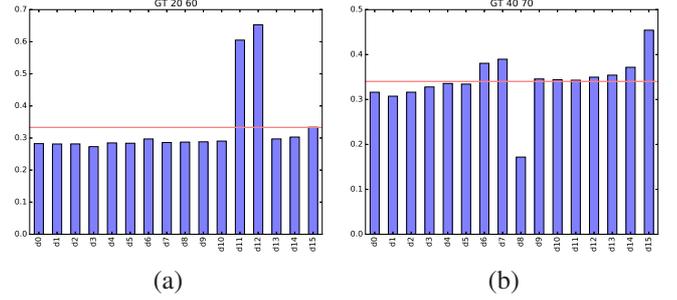


Figure 3: An example of neighboring distances from different time steps of the GT variable of the ionization data set. Every bar shows the distance $d(i, i+1)$. (a) shows one reason for a spike where two neighbors are too far apart. (b) shows a different cause, that is related to a jump discontinuity in the distance function.

318 nearly straight even if the distance functions themselves are not
 319 linear. This allows the desired isovalue to be identified within a
 320 small number of iterations. In our experiment, we find that five
 321 iterations per step is sufficient. For more details, please refer to
 322 Figure 5 and Section 4.1.

3.3. Convergence Stabilization

324 During the refinement stage, some proposed isovalues po-
 325 tentially lead to an adverse change of distance. This typically
 326 translates into distance values, named *spikes*, that are much
 327 higher/lower than the average distance, causing large average
 328 and maximum errors. Figure 3 demonstrates two common types
 329 of spikes: (1) the binary search successfully finds an isovalue
 330 that has mostly equal distances to its two neighbors, but these
 331 distances are larger than the average distance; and (2) the bi-
 332 nary search fails to identify a desired isovalue, and the distance
 333 between this isovalue and one of its neighbors becomes a spike.

334 The first type of spikes between three isovalues v_{i-1}, v_i , and
 335 v_{i+1} can be expressed as $d_{i-1, i} \approx d_{i, i+1}$ and $d_{i-1, i}, d_{i, i+1} \gg d_\mu$.
 336 These spikes usually appear due to the underestimation of the
 337 differences of isosurfaces in the interval $[v_{i-1}, v_{i+1}]$. For ex-
 338 ample, in Figure 3 (a), after one step of the refinement stage,
 339 the binary search identifies an isovalue v_{12} whose distances to
 340 its neighbors (i.e., $d_{11, 12}$ and $d_{12, 13}$) are about twice as high as
 341 the average distance, meaning that the interval between v_{11}, v_{12} ,
 342 and v_{13} may be too large. This type of spikes may gradually dis-
 343 appear since v_{i-1} and v_{i+1} will be moved closer to v_i in the next
 344 step of the refinement stage. For example, since $d_{10, 11}$ is much
 345 smaller than $d_{11, 12}$, v_{11} will be moved closer to v_{12} to reduce
 346 $d_{11, 12}$ for an equal distance between v_{10}, v_{11} , and v_{12} . However,
 347 this type of spikes still causes a steep increase of the average
 348 and maximum errors, leading to an unstable status during the
 349 refinement stage.

350 To alleviate this problem, we propose a *spike treatment* that
 351 rejects isovalues leading to spikes. Formally, for every isovalue
 352 v_i^τ that has been changed in step τ , we compare $d_{i-1, i}^\tau$ and $d_{i, i+1}^\tau$
 353 to d_μ^τ using their relative differences with respect to d_μ^τ .

354 If any of the two difference values surpasses a predefined
 355 spike threshold δ_s , we reject v_i^τ and replace it with $v_i^{\tau-1}$. Note
 356 that the old value $v_i^{\tau-1}$ has more agreeable distances, as v_{i-1}
 357 and v_{i+1} are static in this step. Intuitively, by avoiding the steep

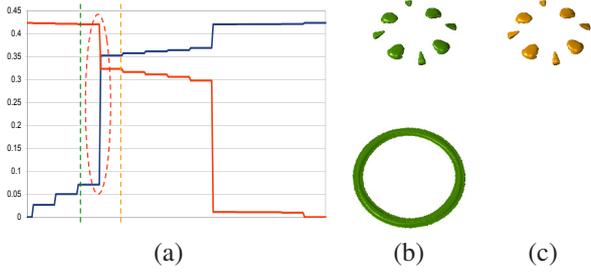


Figure 4: An example of discontinuous distance functions using the GT variable of the ionization data set. (a) shows the distances to isovalue 20547.8 (blue) and to isovalue 20680.4 (red). The horizontal axis represents sampled isovalues and the vertical axis represents the distance. (b) and (c) are the two isosurfaces corresponding to the isovalues highlighted by the dashed lines in (a).

changes, this treatment postpones, instead of preventing, the salient isovalue v_i to be discovered. Therefore, the entire refinement stage will exhibit a more smoothly convergence toward the best solution.

For the second type of spikes, the binary search fails to identify an isovalue v_i with equal distances to its neighbors. In this case, at least one of the distances $d_{i-1,i}$ and $d_{i,i+1}$ will differ from the average distance d_μ . For example, in Figure 3 (b), the distance $d_{8,9}$ is much smaller than the other distances between neighbors, therefore, leading to a larger error. This type of spikes is usually caused by a *jump discontinuity* in the distance functions between the neighboring isovalues.

A jump discontinuity (henceforth jump) appears when the distance function between two isovalues is discontinuous. An example is demonstrated in Figure 4. For the purpose of analysis, we consider two distance functions $d_l(v_i)$ and $d_r(v_i)$, which map an isovalue v_i to its distances to its left and right neighbors, respectively. In this example, we densely sample 200 isovalues between two fixed isovalues 20547.8 and 20680.4, and compute the distance from each sampled isovalue to the fixed ones. In Figure 4 (a), the distances to isovalues 20547.8 and 20680.4 are plotted as blue and red lines, respectively. Unlike the case of two smooth distance functions, as demonstrated in Figure 2 (b), a steep change occurs, highlighted by the red dashed ellipse. This indicates that the two distance functions are not continuous at the corresponding isovalue.

Figure 4 (b) and (c) show the isosurfaces corresponding to the two isovalues on the two sides of this critical isovalue. We can see that this isovalue actually corresponds to the topological change with the lower ring emerging. The blue line appears below (above) the red line before (after) this change. This means, given the properties of the lower and upper bounds, the lower (upper) bound will always be on the left (right) of this isovalue. After several steps, the binary search will be trapped in a small value range centered at this critical isovalue. This does not only lead to a large error by itself but also stops the distance values from propagating from one side of the critical isovalue to the other side. Therefore, when a jump appears, we may only achieve two equal distances on the two sides of the critical isovalue.

To tackle this problem, we propose a *jump treatment* that first identifies the isovalue v^* of the jump and balances the dis-

tances on its two sides. By definition, a jump is a discontinuous point in the distance functions. Therefore, v^* can be detected through examining the following criterion

$$\begin{aligned} d(v_i, v^* + \epsilon) &\gg d(v_i, v^*), \text{ and} \\ d(v^*, v_j) &\gg d(v^* + \epsilon, v_j). \end{aligned} \quad (1)$$

Instead of explicitly detecting v^* , we examine this criterion at each iteration of the refinement stage. Once a jump is encountered, we fix the upper and lower bounds of the binary search so that the jump will reside in the bounded interval, and push the isovalues from one side of the jump to the other side. Let $V_l = \{v_k | 0 \leq k \leq i\}$ and $V_r = \{v_k | j \leq k < n\}$ be the isovalue sets on the left and right sides of the jump, respectively, and let $d_\mu(V_l)$ and $d_\mu(V_r)$ be the average distances of the neighboring isovalues in V_l and V_r , respectively. Without loss of generality, assuming $d_\mu(V_l) < d_\mu(V_r)$, we take an isovalue from V_l and push it to V_r , so that the set of isovalues V_l becomes sparser and the set of isovalues V_r gets denser. This will lead to an increase of $d_\mu(V_l)$ and a decrease of $d_\mu(V_r)$, thus achieving a better balance of the average distances on both sides.

3.4. Distance Measures

We experiment our approach with two different distance measures: the *mean of the closest point distances* (MCP) [7] and the *isosurface similarity map* (ISM) measure [2]. Other distance measures may be applied as well, according to the specific analysis goals.

MCP distance. The MCP distance between two isosurfaces s_i and s_j uses the Euclidean distance to compute the closest distance for every point p_k on s_i to any point p_l on s_j and vice versa. The MCP distance of s_i and s_j is defined as follows

$$\begin{aligned} d_{\text{MCP}}(s_i, s_j) &= \frac{1}{2} (d(s_i, s_j) + d(s_j, s_i)), \text{ where} \\ d(s_i, s_j) &= \frac{\sum_{p_k \in s_i} \min_{p_l \in s_j} \|p_k - p_l\|}{|S_i|}. \end{aligned} \quad (2)$$

ISM distance. The ISM measure inspects the mutual information of the distance fields corresponding to two isosurfaces s_i and s_j . Based on the uniformly sampled distance fields of the two isosurfaces, a joint histogram can be computed to derive the mutual information. Again, we use the Euclidean distance to compute the distance fields for an isosurface s . For each grid point in the distance field, we record two closest distances from that point to the two isosurfaces s_i and s_j , and compute the joint histogram of the distances. The mutual information between two random variables X and Y can be computed from their joint histograms as follows

$$\begin{aligned} I(X, Y) &= H(X) + H(Y) - H(X, Y), \text{ with} \\ H(X) &= - \sum_{x \in X} p_X(x) \log(p_X(x)), \\ H(X, Y) &= - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x, y) \log(p_{X,Y}(x, y)), \end{aligned} \quad (3)$$

where $H(X)$ and $H(Y)$ are the marginal entropies and $H(X, Y)$ denotes the joint entropy of X and Y . In our case, X (Y) is the distance from a grid point in the distance field to isosurface s_i (s_j). We further normalize the mutual information

$$\hat{f}(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)}, \quad (4)$$

and convert the similarity measure into a distance measure

$$d_{\text{ISM}}(s_i, s_j) = 1 - \hat{f}(s_i, s_j). \quad (5)$$

Approximation and acceleration. Distance measures between isosurfaces often share two common steps: constructing isosurfaces and identifying the closest points of given points. When computing the ISM distance, the distance field of an isosurface requires the distance from each grid point to the closest point on the isosurface to be computed. When computing the MCP distance between two isosurfaces, for each point on one isosurface, the closest point on the other isosurface needs to be identified. We take three considerations from [24] to accelerate these two key steps.

First, we approximate each isosurface using a point set instead of extracting the actual surface. Generating the exact isosurface produces multiple points and their connections for each voxel. However, the connections are usually not involved in the distance computation and the points are often unnecessarily dense. This approximation scheme splits the volume into uniform blocks and examines each block to determine whether it contains the isosurface. The centers of blocks that contain the isosurface are considered as an approximation of the isosurface. Using this scheme, the error for computing the closest point is bounded by $(\sqrt{3}/2)l$ (i.e., half the length of a block’s diagonal l).

Second, we build one BVH-tree for each isosurface to organize its approximation points. This allows the closest point on an isosurface to be queried efficiently. For construction efficiency, we use bounding boxes and leverage Karras’ algorithm [23] to build each BVH-tree on GPU in parallel. The BVH-trees are stored in the graphics memory, so that multiple closest point queries can be performed in parallel. In addition, given a point, we estimate the upper bound of the distance to the closest point by uniformly sampling the approximation points. Since the approximation points are loosely ordered following the scanline order, this provides a tighter upper bound and therefore avoids many unnecessary branches of the BVH-tree to be traversed.

Third, both the distance fields and the approximation of isosurfaces can be downsampled to further reduce the time cost. The approximation can be downsampled by scanning blocks of voxels. The centers of blocks that contain the isosurface become the approximation in this case. The error of the closest point is bounded by $(\sqrt{3}/2)l$, where l is the edge length of a block in voxels. For the distance fields, it has been shown that the resolution can be reduced by eight folds along each dimension without sacrificing the quality of the resulting ISMs [2].

Through this acceleration measures, we can achieve a linear time complexity considering all steps to compute neighboring distances, except for building the BVH-trees. The initial approximation examines $O(|V|)$ voxel, where $|V|$ is the size of the volume. For the next step, building the BVH-tree, Karras reported the time complexity of $O(n \log n)$ in the worst case [23]. Note that number of points from the approximation, n , is typically much smaller than $|V|$. Using the BVH-tree, querying the

closest point of a given point takes on average $O(\log n)$ steps. The number of queries for this is bounded by either the size of the distance field ($O(|V|)$) (when using ISM distance) or the size of another surface ($O(n)$) (when using MCP distance). Using the GPU, multiple queries can be performed in parallel. For the ISM distance, we further compute mutual information, by examining every point in the distance field. In our experiment, we find that computing the distance between two isosurfaces already fully utilizes the computation power of a single GPU. Therefore, the cost of our approach is linear to the number of isosurfaces if a single GPU is used. Multiple GPUs, if available, can be readily utilized as computing multiple distances is embarrassingly parallel.

4. Results

We mainly run our experiments on a desktop with an Intel Core i7-4790 quad-core CPU @ 3.6 GHz, 32 GB RAM, and an NVIDIA GeForce GTX 760 GPU accelerator. For further exploration of time-varying data sets, we leveraged a cluster with a shared GPU queue. The shared GPU queue uses the following systems:

- 8 Quantum TXR231-1000R servers with dual Intel Xeon 12-core CPU E5-2650 v4 @ 2.20GHz, 128 GB RAM, and 4 NVIDIA TITAN X (Pascal) GPU accelerators;
- 8 Quantum TXR231-1000R servers with dual Intel Xeon 12-core CPU E5-2650 v4 @ 2.20GHz, 128 GB RAM, and 4 NVIDIA Tesla P100-PCIE-16GB GPU accelerators.

The queue distributes the workload on different machines depending on the availability. Since we were only interested in the number of iteration needed to achieve a good solution, we did not restrict our runs to a single hardware configuration. In the following, we first analyze our general approach quantitatively (Section 4.1) and qualitatively (Section 4.2), and then study the impact of the spike treatment and jump treatment (Section 4.3).

4.1. Quantitative Study

Quality measures. We evaluate the quality of a set of selected isovalues $V = \{v_1, \dots, v_n\}$ based on the distances among neighbors (i.e., $d_{1,2}, \dots, d_{n-1,n}$) and the average distance d_μ . For each distance $d_{i-1,i}$, we compute an error term $e_{i-1,i}$ to indicate the difference between this distance and the average distance

$$e_{i-1,i} = \frac{\|d_{i-1,i} - d_\mu\|}{d_\mu}, \quad (6)$$

where dividing the absolute difference by the average distance normalizes the error term. In this paper, we quantify the quality of selected isovalues using the average error

$$e_\mu = \frac{\sum_{i=2}^n e_{i-1,i}}{n-1}, \quad (7)$$

and the maximum error

$$e_{\max} = \max_{2 \leq i \leq n} \{e_{i-1,i}\}. \quad (8)$$

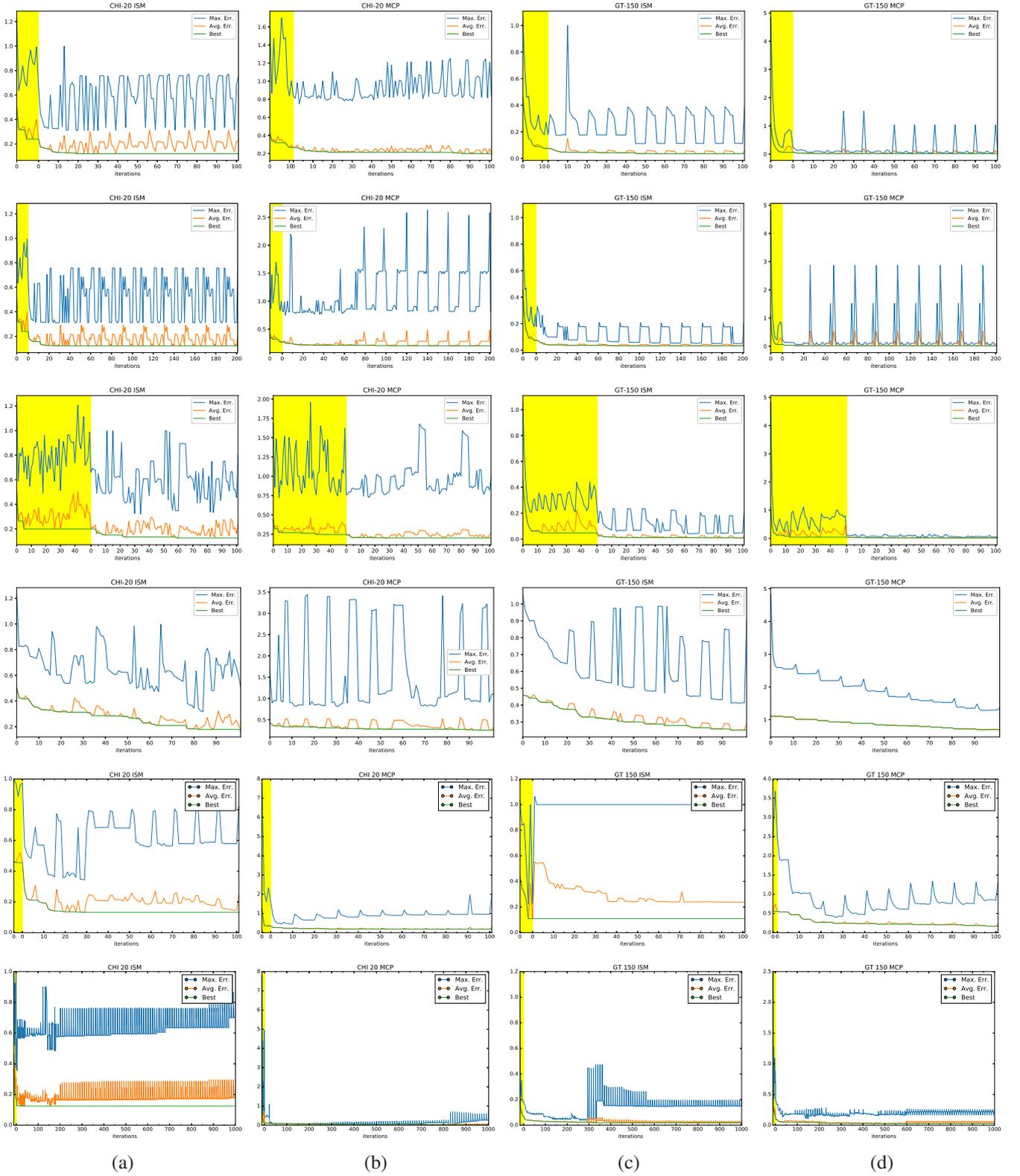


Figure 5: Parameter choices on error curves. (a) and (b) show results using the scalar dissipation rate (CHI) variable of the combustion data set at time step 20. (c) and (d) show results using the gas temperature (GT) variable of the ionization data set at time step 150. (a) and (c) show results using the ISM distance, and (b) and (d) show results using the MCP distance. Rows from top to bottom show results with $\langle \delta_e, \delta_r \rangle = \langle 10, 5, 20 \rangle$, $\langle 10, 10, 20 \rangle$, $\langle 50, 5, 20 \rangle$, $\langle 0.5, 20 \rangle$, $\langle 1, 5, 20 \rangle$, and $\langle 10, 5, 200 \rangle$, respectively. In each plot, the yellow and white background colors indicate the estimation and refinement stages, respectively. The blue, orange, and green curves show the maximum error, average error, and best error over iterations.

data set	dimension x, y, z, v, t	avg. # iterations		timing (sec.)		average error			refine improve	difference to best (%)			
		estimate	refine	estimate	refine	initial	estimate	best		20 iter.	40 iter.	60 iter.	80 iter.
atmosphere	147, 129, 49, 4, 121	11.1	57.9	3.36	27.61	0.437	0.153	0.081	47.29%	26.62	13.47	5.71	1.71
climate	360, 66, 27, 2, 120	11.8	47.5	2.87	22.21	1.060	0.251	0.165	34.12%	10.12	6.37	1.32	0.07
combustion	480, 720, 120, 5, 122	11.9	68.8	27.90	211.35	0.417	0.184	0.099	46.20%	25.39	13.16	5.31	1.16
hurricane	500, 500, 100, 11, 48	11.2	59.2	28.07	230.11	0.427	0.150	0.073	51.07%	40.50	18.08	7.65	3.92
ionization	600, 248, 248, 8, 199	11.5	63.9	20.70	164.63	0.597	0.287	0.188	34.33%	30.12	20.49	10.85	2.79
vortex	128, 128, 128, 1, 90	13.3	95.0	8.49	60.73	0.158	0.041	0.011	73.81%	41.42	28.95	20.22	15.27

(a)

data set	dimension x, y, z, v, t	avg. # iterations		timing (sec.)		average error			refine improve	difference to best (%)			
		estimate	refine	estimate	refine	initial	estimate	best		20 iter.	40 iter.	60 iter.	80 iter.
atmosphere	147, 129, 49, 4, 121	11.1	59.6	4.10	31.70	0.525	0.198	0.099	49.94%	57.60	19.91	8.92	2.74
climate	360, 66, 27, 2, 120	12.3	93.8	6.34	48.29	0.863	0.260	0.033	87.15%	206.16	104.33	53.49	27.29
combustion	480, 720, 120, 5, 122	12.9	72.2	91.51	646.55	0.560	0.205	0.099	51.54%	46.03	27.18	16.89	9.50
hurricane	500, 500, 100, 11, 48	12.8	88.8	49.57	379.31	0.281	0.061	0.010	82.17%	98.95	49.67	28.76	11.92
ionization	600, 248, 248, 8, 199	11.8	62.3	38.65	290.46	0.524	0.209	0.135	34.98%	27.79	15.26	6.08	2.72
vortex	128, 128, 128, 1, 90	11.0	80.0	15.42	126.15	0.533	0.194	0.072	62.81%	61.68	46.20	28.25	14.36

(b)

Table 1: Performances using (a) the ISM distance and (b) the MCP distance. The three columns of average errors show the initial average error of uniform sampling, the average error after the estimation stage, and the average error of the best solution. The column “refine improve” shows the percentage of average error reduced by the refinement stage. The four columns of “difference to best” show the percentage of difference between the average error of the best solution and the average errors after 20, 40, 60, and 80 iterations in the refinement stage.

Since the maximum error is usually determined by the nature of the data sets and the distance measures, as will be shown in Section 4.3, we focus on the average error and use it to determine the best solution, i.e., the set of isovalues with the minimum average error. We do not use the variation or standard deviation to evaluate whether the distances are similar since both of them are dominated by the maximum error when the other errors are small.

Parameter choices. Our approach has three parameters: δ_e the minimum number of iterations in the estimation stage, δ_τ the number of iterations at each step in the refinement stage, and δ_r the number of steps in the refinement stage. For simplicity, we use a 3-tuple $\langle \delta_e, \delta_\tau, \delta_r \rangle$ to denote a parameter setting. Figure 5 shows the results of using two variables of the combustion and ionization data sets for both the ISM and MCP distance measures with six different sets of parameter values ($\langle \delta_e, \delta_\tau, \delta_r \rangle = \langle 10, 5, 20 \rangle$, $\langle 10, 10, 20 \rangle$, $\langle 50, 5, 20 \rangle$, $\langle 0, 5, 20 \rangle$, $\langle 1, 5, 20 \rangle$, and $\langle 10, 5, 200 \rangle$). For each run, we plot the maximum error, the average error, and the current best solution over iterations. The current best solution is the one with the minimum average error obtained up to the current iteration.

We first investigate the impact of parameter δ_τ . In the top two rows of Figure 5, we fix the two parameters $\delta_e = 10$ and $\delta_r = 20$ and compare the performance of $\delta_\tau = 5$ (first row) and $\delta_\tau = 10$ (second row). At each step, having more iterations may potentially allow better convergence of the binary searches. But overall, we do not see a noticeable improvement of accuracy using $\delta_\tau = 10$ over $\delta_\tau = 5$ since the shape of the green curves (best solution) in the same column are mostly the same. With the same number of steps ($\delta_r = 20$), this indicates that we obtain similar results using $\delta_\tau = 10$ but with twice the number of iterations as using $\delta_\tau = 5$.

Next, we study the impact of parameter δ_e . In the first and third rows of Figure 5, we use $\delta_e = 10$ and $\delta_e = 50$, respectively. The other two parameters are fixed ($\delta_\tau = 5$ and $\delta_r = 20$). We find that more than ten iterations in the estimation stage are usually unnecessary since the best solution is mostly unchanged after ten iterations, as shown in the third row of Figure 5. In

addition, we do not find that having more iterations in the estimation stage helps the refinement stage reach the best solution faster. The green curves in the white background, corresponding to the best solution in the refinement stage, demonstrate similar decreasing patterns.

However, we still find that the estimation stage is necessary for the refinement stage to quickly reach its best solution. In the fourth row of Figure 5, we experiment our approach with only the refinement stage, i.e., $\delta_e = 0$. With this setting, we find that the refinement stage approaches the optimal solution much slower. For example, using CHI of the combustion data set and the ISM distance measure shown in (a), the best solution slowly improves over the 100 iterations without the estimation stage and reaches the minimum average error of 0.179 at the last iteration. With the estimation stage, the best solution until the 20-th iteration in the refinement stage has an average error of 0.121, which is already very close to the minimum average error of 0.120 for the entire 100 iterations. The computation time of each iteration in the estimation and refinement stages is similar, since both of them are dominated by the computation of distances between neighbors. Therefore, including the estimation stage clearly gives a better performance.

The fifth row shows the results with only one iteration in the estimation stage, i.e., $\delta_e = 1$. In contrast to setting $\delta_e = 10$, letting $\delta_e = 1$ leads to a more stable convergence, implying that the algorithm got stuck in a local optimum. Note that for the CHI variable at time step 20, we sometimes obtain empty isosurfaces. In this case, we run the estimation stage for more iterations until we obtain a set without empty surfaces.

In the last row, we show the results with $\langle \delta_e, \delta_\tau, \delta_r \rangle = \langle 10, 5, 200 \rangle$ (1000 total iterations in the refinement stage). For some data sets, we achieve a slightly better solution several hundreds iterations later than the best solution achieved within 100 iterations. However, the overall convergence pattern does not change. We believe that, given the time-quality trade-off, a relatively good solution can be found within 100 iterations.

In Figure 6, we show the visual differences among three sets of isosurfaces identified in 20, 100, and 1000 iterations, respectively, using the GT variable of the ionization data set at time

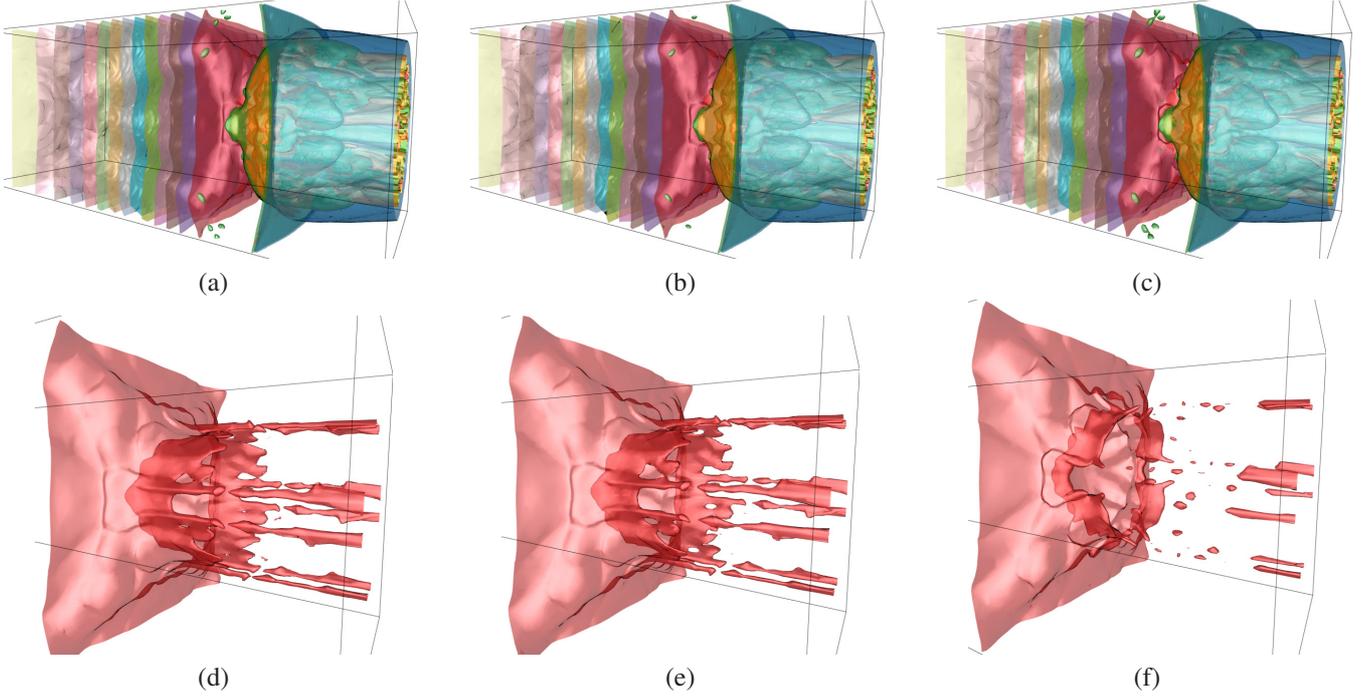


Figure 6: Comparison of the isosurfaces identified in different numbers of iterations using the GT variable of the ionization data set at time step 150. The isosurfaces are selected from the best solutions after 20 ((a) and (d)), 100 ((b) and (e)), and 1000 ((c) and (f)) iterations in the refinement stage. The top row shows 17 selected isosurfaces rendered together. The bottom row shows a single surface highlighting the fine differences.

625 step 150. While the best solution after 1000 iterations is found
 626 at iteration 333 with an average normalized error of 1.86%, the
 627 earlier ones 2.89% (at iteration 92) and 3.92% (at iteration 20)
 628 show small errors as well. Although the relative difference in
 629 average error seems enormous, their absolute difference is still
 630 small. Visual comparison confirms that similar isosurfaces are
 631 identified. In the top row of Figure 6, we show all isosurfaces
 632 corresponding to solutions at iteration 20 (a), 92 (b), and 333
 633 (c), respectively. We find that the overall difference is barely
 634 visible. Therefore, to inspect more closely, we depict the fourth
 635 isosurface of the selected sets in the second row. We can clearly
 636 see a difference between the first two images ((d) and (e)) and
 637 the last one (f). Furthermore, subtle differences between (d) and
 638 (e) can be seen. Although single surfaces differ among the dif-
 639 ferent solution sets, the overall sets look fairly similar, offering
 640 a comprehensive overview of the volumetric data set.

641 In our experiment, we use a large enough value of $\delta_r = 20$
 642 to study how the best solution evolves over iterations. We find
 643 that the setting of $\langle \delta_e, \delta_\tau, \delta_r \rangle = \langle 10, 5, 20 \rangle$ usually yields
 644 good results in terms of timing and error. Therefore, we use
 645 this setting for reporting the remaining results.

646 **Timing and quality.** As shown in Table 1, we study the
 647 timing and quality performance using six data sets with differ-
 648 ent characteristics. For each data set, we use all the associated
 649 variables with three time steps selected (the beginning, middle,
 650 and ending time steps). Collected for each variable and each
 651 selected time step, the results are averaged for each data set.

652 Table 1 (a) shows the results using the ISM distance. Our
 653 approach produces mostly equally spaced isosurfaces with the
 654 average errors smaller than 0.1 for most of the data sets except

655 the climate (0.165) and ionization (0.188) data sets. This may
 656 be related to the structures of the data sets as their initial er-
 657 rors are the largest among these six data sets. The number of
 658 iterations in the estimation stage is stable for all data sets and
 659 slightly above our minimum number ($\delta_e = 10$). In the refine-
 660 ment stage, our approach reaches the best solution around 60
 661 iterations for most of the data sets, except the vortex data set
 662 (averaging 95 iterations). The climate data set even reaches so-
 663 lutions that are close to the best solution within 20 iterations,
 664 with only 10.12% difference. The other data sets except the
 665 vortex data set have differences less than or around 20% within
 666 40 iterations and less than or around 10% within 60 iterations.
 667 Although having the slowest convergence, the vortex data set
 668 achieves the smallest average error (0.011) among all data sets
 669 after the refinement stage. In Figure 9 (k) and (l), we find that
 670 the green curve declines slowly after 15 iterations in the re-
 671 finement stage. The higher percentages shown in the table are
 672 probably due to the small average error. The average errors are
 673 smaller than 0.3 for all the data sets after the estimation stage,
 674 and the refinement stage further reduces the average errors by at
 675 least 30%, which confirms the necessity of the refinement stage.
 676 Our approach performs efficiently using the ISM distance. To
 677 process one volume, it takes around one minute for the atmo-
 678 sphere, climate, and vortex data sets, and less than five minutes
 679 for the other data sets.

680 Table 1 (b) shows the results using the MCP distance. In
 681 general, we find that it takes more iterations for the refine-
 682 ment stage to approach the best solutions using the MCP dis-
 683 tance. Three data sets reach the best solutions after 80 iter-
 684 ations. Within 40 iterations, only two data sets obtain good

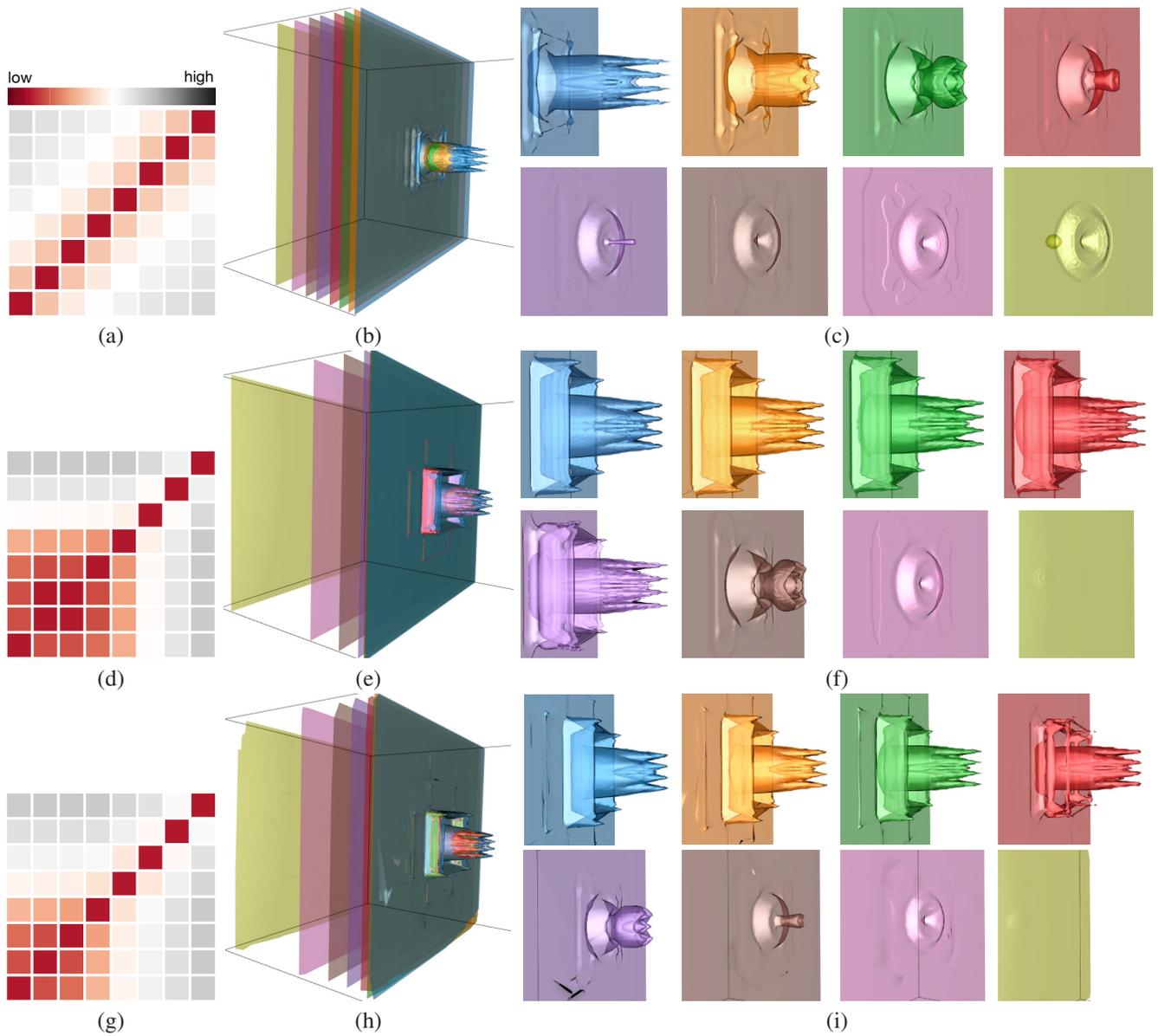


Figure 7: Comparison of our approach (top row), the ISM approach (middle row), and the k-means approach (bottom row) using the GT variable of the ionization data set at time step 10. (a), (d), and (g) show the distance matrices of the selected isosurfaces. Note that the distances are normalized and all matrices use the same color map. (b), (e), and (h) show all the selected isosurfaces in a single image. (c), (f), and (i) show the central regions of individual isosurfaces in separate images.

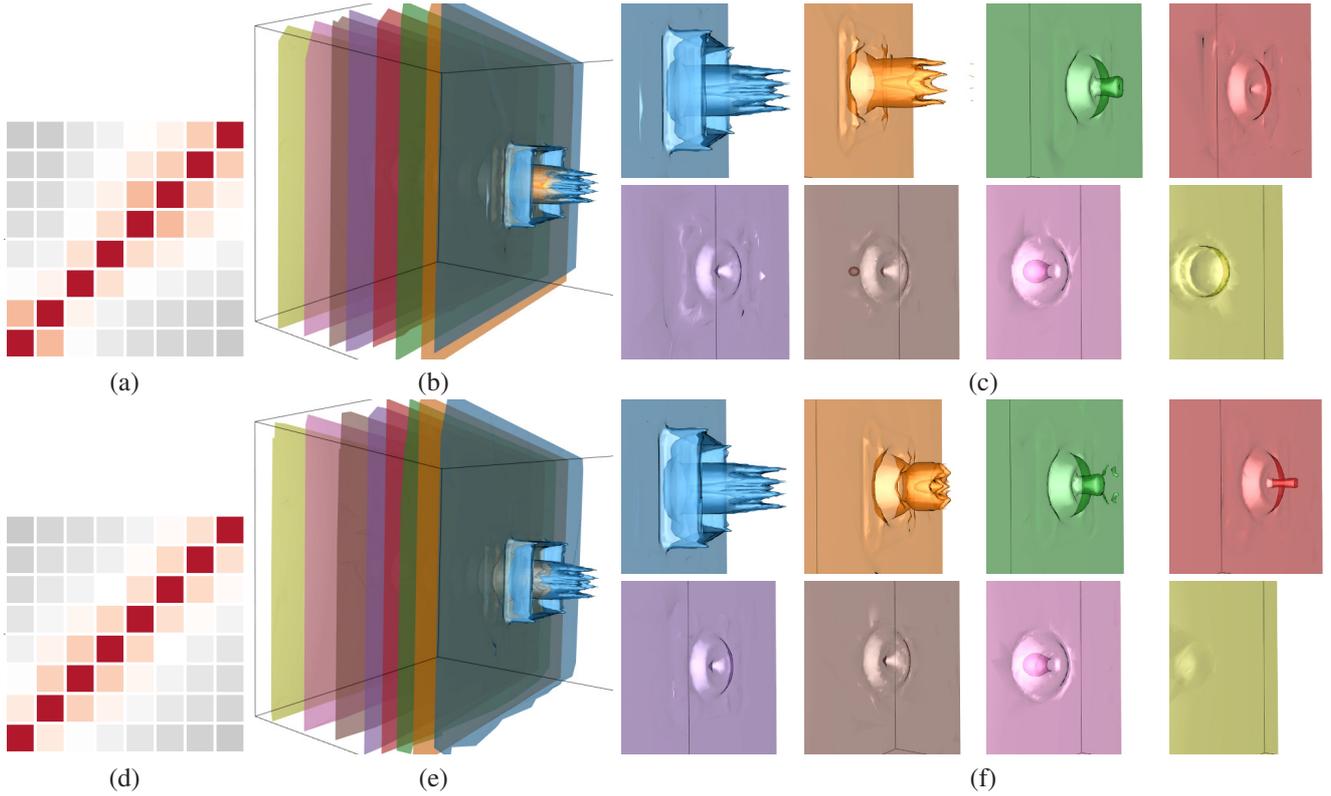


Figure 8: Comparison of results using a set of 128 nearly equally spaced isosurfaces as input for the ISM approach (top row) and the k-means approach (bottom row) using the GT variable of the ionization data set at time step 10. (a) and (d) show the distance matrices, (b) and (e) show the representative isosurfaces rendered together, while (c) and (f) show the individual surfaces.

685 solutions whose average errors have less than 10% differences
 686 from the best solutions. For five of the data sets, it takes 80 itera-
 687 tions to reach reasonably good solutions with less than 15%
 688 difference from the best solutions. However, we find that the
 689 average errors are usually smaller using the MCP distance. The
 690 largest average error is 0.135 using the ionization data set, and
 691 all the other data sets have average errors smaller than 0.1. The
 692 refinement stage provides more significant improvement using
 693 the MCP distance. It reduces the average error by at least 80%
 694 for two data sets and 49% for five data sets. Actually, we find
 695 that the data sets with a smaller best error usually benefit more
 696 from the refinement stage and take more iterations to reach the
 697 best solution. This is likely related to the intrinsic structures of
 698 the data sets. The distance functions are probably more contin-
 699 uous using these data sets so that the binary search in the
 700 refinement stage is less likely to be trapped by the discontinu-
 701 ous points in the distance function. Although the average error
 702 is smaller, we find that the MCP distance takes more time
 703 to compute. The combustion data set requires the longest total
 704 computation time of around nine minutes. The computation
 705 time for the other data sets varies from one to seven minutes.

706 4.2. Qualitative Study

707 **Comparison to other approaches.** For visual comparison,
 708 we generate eight isosurfaces of the GT variable of the ioniza-
 709 tion data set at time step 10 using our approach, the ISM ap-
 710 proach [2], and k-means clustering [6]. For our approach, we
 711 fix the minimum and maximum isovalues and compute eight

712 isovalues between them. For the other approaches, we evenly
 713 sample 128 isovalues and identify eight representative ones.
 714 We use Imre et al.’s [24] approximation to compute the ISM
 715 distances between the sampled isovalues for all three variants.
 716 The implementation of the ISM algorithm is based on the priori-
 717 ty queue algorithm [2] and the k-means clustering is based on
 718 Lloyd’s algorithm [25]. Figure 7 (a) shows the distance matrix
 719 of the isosurfaces selected by our approach. Note that in this,
 720 and the following images, the distance values are in $[0, 1]$. We
 721 can see that the cells recording the differences between neigh-
 722 boring isosurfaces (i.e., the cells that are next to the diagonal
 723 cells) share similar colors. This indicates similar distances be-
 724 tween neighbors. Figure 7 (b) confirms this observation as the
 725 selected isosurfaces distribute evenly in the space. The eight
 726 isosurfaces demonstrate a smooth transition of the features at
 727 the center of each isosurface, as shown in Figure 7 (c).

728 In contrast, five of the representative isosurfaces selected by
 729 the ISM approach and four of the representative isosurfaces se-
 730 lected by the k-means approach are similar. In Figure 7 (d), we
 731 can see a 5×5 block at the bottom left corner of the distance
 732 matrix of the representative isosurfaces, indicating high similar-
 733 ities among the corresponding isosurfaces. Similarly, Figure 7
 734 (h) contains a 4×4 block.

735 In Figure 7 (e), we can see that the five similar representa-
 736 tive isosurfaces collapse in space. Therefore, five of the feature
 737 regions in the representative isosurfaces actually corresponds
 738 to the nearly identical structure, as shown in Figure 7 (f). Al-

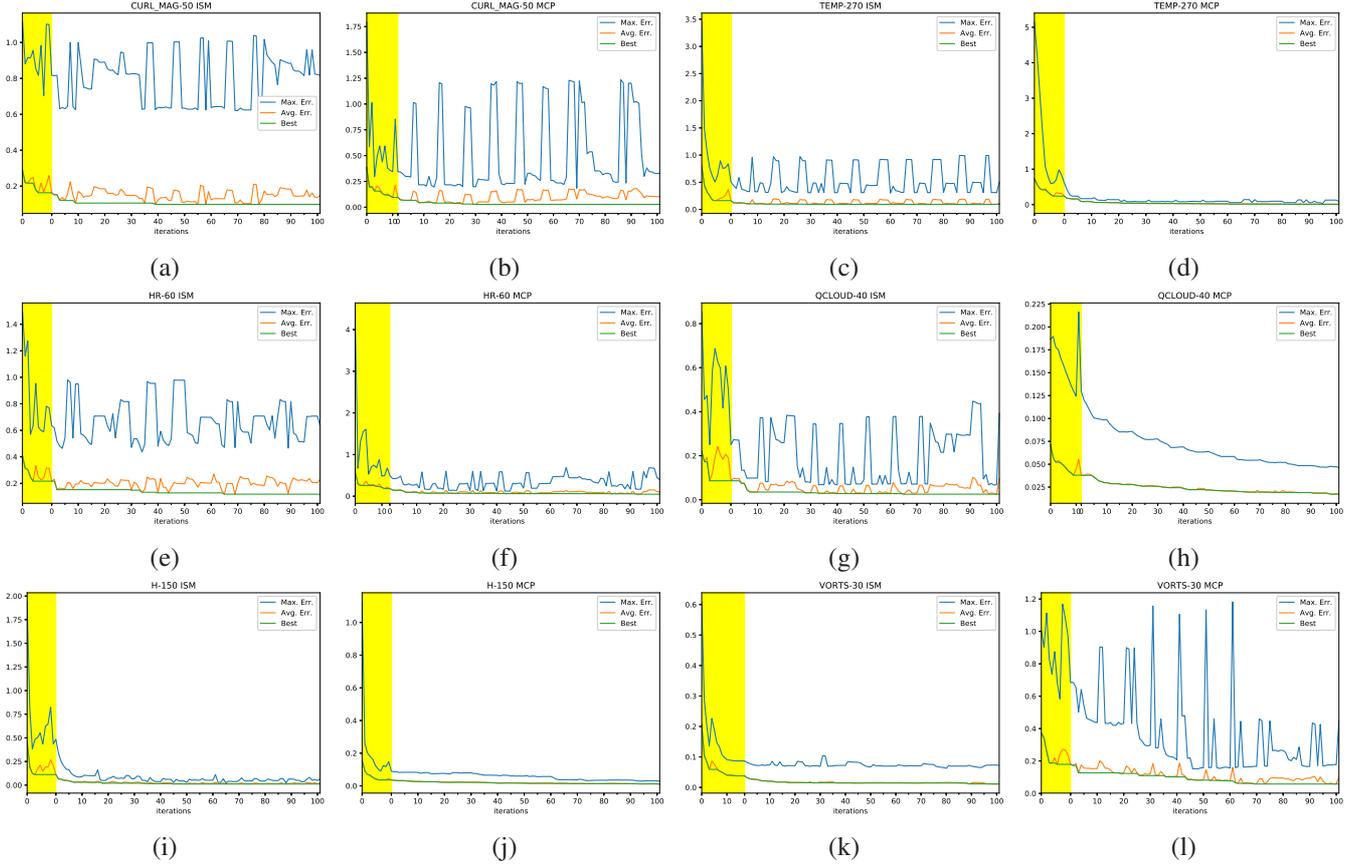


Figure 9: Typical error curves over iterations using the ISM distance (first and third column) and the MCP distance (second and fourth). (a) and (b) show the results of the curl magnitude (CURL_MAG) variable of the atmosphere data set at time step 50. (c) and (d) show the results of the temperature (TEMP) variable of the climate data set at time step 270. (e) and (f) show the results of the heat release (HR) variable of the combustion data set at time step 60. (g) and (h) show the results of the cloud moisture mixing ratio (QCLOUD) variable of the hurricane data set at time step 40. (i) and (j) show the results of the H mass abundance (H) variable of the ionization data set at time step 150. (k) and (l) show the results of the vortex data set at time step 30.

739 though the ISM approach has a scheme to prevent similar isosurfaces to be selected [2], this scheme may be ineffective when
 740 the input is biased. As a matter of fact, more than half of the sampled isosurfaces in this volume correspond to the same
 741 structure. In general, we find that the distance matrix of the representative isosurfaces identified by the ISM approach often
 742 exhibits this kind of blocking effect for the structures captured by more sampled isosurfaces. This echoes that it is critical to
 743 producing unbiased isosurfaces as input for the surface-based volumetric data analysis algorithms. In addition, Figure 7 (h)
 744 shows the same effect for the k-means clustering with four similar isosurfaces selected. In Figure 7 (i), a closer inspection
 745 reveals that three of the four surfaces are very similar, with the fourth one being closely related to them.

753 In contrast, Figure 8 depicts the representative isosurfaces selected by the ISM [2] and k-means clustering [6] approaches
 754 using 128 nearly equally spaced isosurfaces as input instead of the uniformly sampled ones. We use the GT variable of the
 755 ionization data set at time step 10. In (a) and (d), the distance matrices do not show the strong blocking effect, which means
 756 that the problem of oversampling certain value ranges could be circumvented. Compared to Figure 7, we can see that there is
 757 a shift between the representative surfaces selected, allowing to
 758

762 further explore previously overseen isosurfaces. For example, the third surface in the second row (pink) in both (c) and (f)
 763 has not been discovered previously. This further indicates that an unbiased input may improve the understanding of the underlying
 764 surfaces.

767 **Comparison of ISM and MCP distances.** For a qualitative study of the impact of distance measures, we first investigate the
 768 error curves using the ISM distance and the MCP distance, as shown in Figure 9. For each volume, we chose to identify 15
 769 equally spaced isovalues between the minimum and maximum isovalues. In general, the curves confirm our finding in Table 1
 770 that the MCP distance has a slightly smaller average error. The only exception is the vortex data set. Figure 9 (k) and (l) show
 771 that the MCP distance converges slower with unstable spikes of the maximum error curve for this data set. Figure 10 shows the
 772 distance matrix and a set of selected isosurfaces using the vortex data set for each distance measure. In Figure 10 (a), we can
 773 see that the distances between neighbors are actually similar for both measures. However, the two distance measures behave
 774 differently with this data set: the MCP distance changes in a smoother manner when the isovalues become more different,
 775 while the ISM distance seems to better distinguish isovalues in a smaller value range. For each distance measure, we evenly
 776

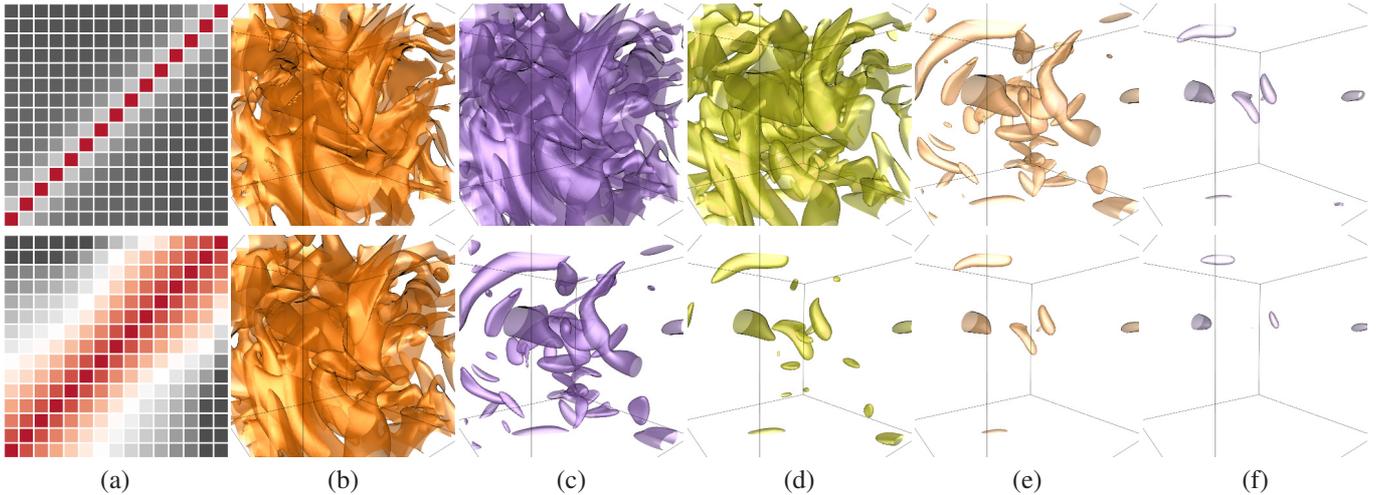


Figure 10: Comparison of the ISM distance (top row) and the MCP distance (bottom row) using the vortex data set at time step 30. (a) shows the distance matrices of the selected isosurfaces. (b) to (f) show the two sets of isosurfaces chosen by the respective distance measures.

785 select five isosurfaces (the second, fifth, eighth, 11th, and 14th)
 786 from the set of fifteen isosurfaces, as shown in Figure 10 (b) to
 787 (f). We find that the ISM distance identifies more large-scale
 788 isosurfaces while the MCP distance selects more small-scale
 789 isosurfaces. This is probably because the ISM distance better
 790 distinguishes the large isosurfaces and the MCP distance better
 791 differentiates the small ones using this data set.

792 We then investigate the combustion data set, for which the
 793 average errors are similar using both measures. We show the
 794 results of the HR variable in the top two rows of Figure 11. For
 795 this variable, although both the error curves in Figure 9 (e) and
 796 (f) and the distance matrices in Figure 11 (a) indicate smaller
 797 errors using the MCP distance, we find that the isosurfaces se-
 798 lected using the two distance measures are actually similar, as
 799 shown in the top two rows of Figure 11 (b) to (d). The isosur-
 800 faces are the ninth, 12th, and 14th from the fifteen selected ones.
 801 The first nine isosurfaces all demonstrate small-scale structures,
 802 which are visually similar. For the YOH variable, the distance
 803 matrices and the fourth, eighth, 12th isosurfaces (evenly sam-
 804 pled) are shown in the bottom two rows of Figure 11. We can
 805 see that the ISM distance selects more small-scale isosurfaces,
 806 which contradicts our findings with the vortex data set. We ac-
 807 tually find that both measures are sensitive to changes on small-
 808 scale isosurfaces in general. This leads to the conclusion: the
 809 behavior of the two distance measures heavily depends on the
 810 spatial distribution of the isosurfaces. While large isosurfaces
 811 usually have stable spatial distributions, changes (even if they
 812 are tiny) on small isosurfaces may lead to significantly differ-
 813 ent spatial distributions. Therefore, due to the nature of these
 814 two measures, the differences among small isosurfaces are of-
 815 ten emphasized.

816 4.3. Discussion

817 **Time-varying data sets.** We further experiment possible
 818 solutions to improve the performance of time-varying data sets.
 819 In Section 4.1, we demonstrate that the refinement stage con-
 820 verges much faster with the use of the estimation stage, which

data set	# vol.	better vol. (%)		# iter. (%)	
		best	10% diff.	best	10% diff.
atmosphere	242	87.6	91.3	57.2	38.7
climate	151	72.9	99.3	91.4	35.20
combustion	61	0.0	1.6	N/A	18.93
hurricane	19	79.0	84.2	42.6	33.7
ionization	25	16.0	24.0	57.3	57.2
vortex	45	77.8	82.2	21.0	18.4

Table 2: Using the isovalues selected at the previous time step as initial isovalues for the next time step. “# vol.” shows the number of volumes experimented with. “better vol. (%)” shows the percentage of volumes achieving better results. “# iter. (%)” shows the percentage of iterations spent to achieve better results. “best” indicates that the result is better than the best solution obtained from the normal run, and “10% diff” indicates that the result is within 10% difference from the best solution of the normal run.

821 indicates the importance of a good initial set of isovalues. Ob-
 822 serving that the structures of volumes usually change gradually
 823 over the time steps, we hypothesize that using the isovalues se-
 824 lected for the same variable at the previous time step will speed
 825 up the computation. Although this strategy has not been fully
 826 studied, we discuss some preliminary findings. We use a clus-
 827 ter with a shared GPU queue to experiment with the six data
 828 sets shown in Table 2. For each variable, we have a normal
 829 run that starts from the uniform sampling of the value range for
 830 each time step, and a run that starts from the isovalues selected
 831 at the previous time step. Each computation node in the GPU
 832 queue performs one run of a variable. Since the cluster contains
 833 computation nodes of different configurations, we compare the
 834 performance using the number of iterations instead of the com-
 835 putation time.

836 Our experiment shows that we do not always get better re-
 837 sults by starting from the isovalues at the previous time step.
 838 Keeping the isovalues produces better results than the normal
 839 run for more than 70% of the volumes with four data sets. How-
 840 ever, for the combustion and ionization data sets, this strategy
 841 fails to produce better or even similar results. For the volumes
 842 that better results are obtained, it generally takes much fewer it-

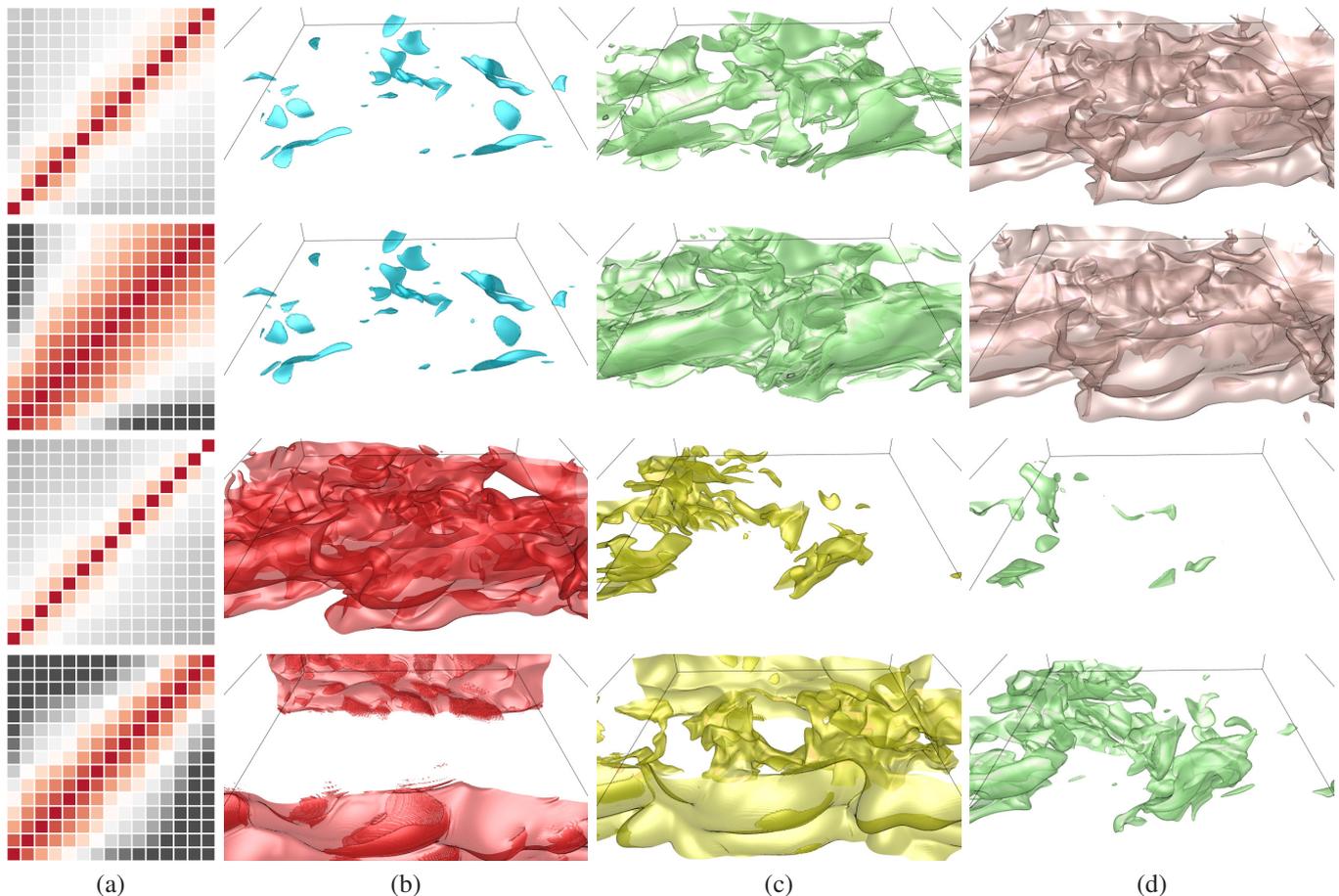


Figure 11: Comparison of the ISM distance and the MCP distance using the heat release (HR) and hydroxyl radical mass fraction (YOH) variables of the combustion data set at time step 60. Rows from top to bottom show the results using HR with ISM distance, HR with MCP distance, YOH with ISM distance, and YOH with MCP distance, respectively. (a) shows the distance matrices of the selected isosurfaces. (b) to (d) show the four sets of isosurfaces chosen by the respective combinations of variable and distance measure.

843 erations to achieve the better results (less than 60% for three of
 844 the four data sets). For the vortex data set, it only takes 21% of
 845 the number of iterations compared to the normal run. For five
 846 of the data sets, it takes less than 40% of the number of iterations
 847 to achieve a similar result. However, the conditions under
 848 which this strategy will perform effectively are still not clear.
 849 We further investigate the impact of the overlap percentage of
 850 the value ranges at neighboring time steps and the average error
 851 at the previous time step, but none of them exhibits a significant
 852 impact on the performance. It seems that the performance
 853 of this strategy heavily relies on the nature of the data since for
 854 all the data sets shown in Table 2, the percentage of volumes
 855 with better results is either higher than 70% or less than 20%.
 856 If we can determine in advance that a time-varying data set is
 857 suitable for this strategy, nearly $2\times$ speedup can be obtained.

858 **Spike treatment.** To analyze the impact of the parameter
 859 δ_s , we conduct an experiment using the common setting of \langle
 860 $\delta_e, \delta_\tau, \delta_r \rangle = \langle 10, 5, 20 \rangle$ but varying δ_s from 0.05 to 0.25 in
 861 steps of 0.05. Note that the original approach without spike
 862 treatment can be considered as setting $\delta_s = \infty$, meaning that all
 863 spikes are tolerated and will not be explicitly treated. Figure 12
 864 shows the results for the GT variable of the ionization data set
 865 at time step 60. The figure represents typical error curves for

866 the ISM distance measure with different δ_s settings. We can
 867 see that using a high tolerance value for δ_s , as seen in (a) the
 868 original ($\delta_s = \infty$), (e) $\delta_s = 0.2$, and (f) $\delta_s = 0.25$, yields high
 869 spikes in the maximum error, resulting in spikes in the average
 870 error. Dampening those instabilities by reducing the threshold
 871 translates to fewer negative changes as can be seen in (b) $\delta_s =$
 872 0.05, (c) $\delta_s = 0.1$, and (d) $\delta_s = 0.15$. This further allows us
 873 to achieve a lower average error for the two variations shown
 874 in (c) and (d). Using $\delta_s = 0.05$ may easily get trapped in a
 875 local optimum, since this parameter setting is too strict to allow
 876 any drastic changes that could resolve the problem. Setting δ_s
 877 to 0.1 or 0.15 leads to the most stable convergence, showing
 878 that those values offer a good balance between allowing too
 879 little changes ($\delta_s = 0.05$) and allowing too much changes ($\delta_s \geq$
 880 0.2). However, the best average errors achieved by setting $\delta_s =$
 881 0.1, 0.15, 0.2, 0.25 are similar. We will study the impact of δ_s to
 882 the best average errors using more data sets later in this section.

883 **Jump treatment.** To analyze the impact of jumps and our
 884 treatment, we experiment our approach with several configura-
 885 tions (with and without spike treatment) using the GT variable
 886 of the ionization data set at time step 20. The error curves are
 887 shown in Figure 13. In (a), we can see the original without jump
 888 or spike treatments. The rest shows the different parameter set-

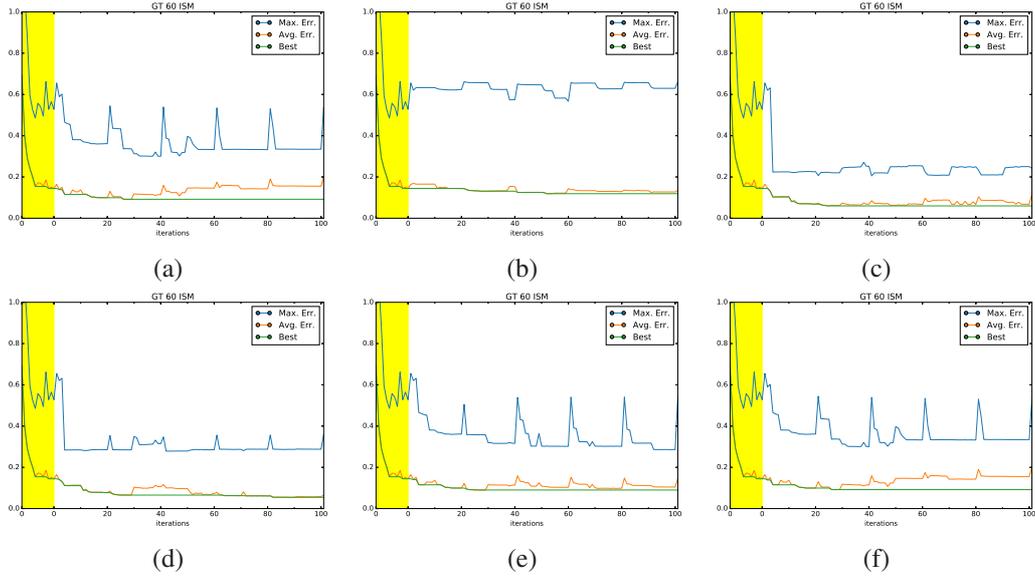


Figure 12: Comparison of the impact of δ_s on the convergence and the final result using the ISM distance on the GT variable of the ionization data set at time step 60. (a) shows the original error curve over the set of iterations (i.e., $\delta_s = \infty$). (b) to (f) show the configurations with $\delta_s = \{0.05, 0.1, 0.15, 0.2, 0.25\}$, respectively.

889 tings for δ_s to treat spikes while handling jumps at the same
890 time. In our experiments, we encounter at most three jumps
891 during the refinement stage using this data set. In (c) to (e),
892 a strict choice of $\delta_s = 0.05, 0.1, 0.15$ restricts isovalue changes
893 to a degree where it is not possible to detect a single jump in
894 the distance functions. By allowing more drastic changes to
895 the isovalues ((f) $\delta_s = 0.20$, (g) $\delta_s = 0.25$, and (b) $\delta_s = \infty$), we
896 can see that jumps appear around strong fluctuation on the error
897 curve. Treating these jumps helps to reduce the fluctuation
898 from that point on. In (b), although the error curve still has
899 strong spikes after treating the first jump, it finds a better so-
900 lution than the original approach in a later iteration. An even
901 stronger reduction in fluctuation of the error curve can be seen
902 in (f). Dampened by spike treatment, the first jump is detected
903 at iteration 70 and its treatment allows the curve to come down
904 further, achieving the best solution for this data set at iteration
905 90. However, jump treatment does not always lead to better
906 convergence or lower average and maximum error values. Fig-
907 ure 13 (g) showcases this. Instantly after the jump around iter-
908 ation 50, the error function cannot recover, resulting in another
909 jump quickly. This results in multiple fixed isovalues, allowing
910 fewer changes in the set of isovalues at every iteration, which
911 hinders the optimization process.

912 As mentioned previously, it is not always clear beforehand
913 whether or not jump treatment improves the best solution due to
914 multiple reasons. First, we treat each jump when it is detected at
915 the refinement stage without the knowledge of all jumps. There-
916 fore, treating one jump may lead to the appearance of an undis-
917 covered jump. As shown in Figure 13 (g), the treatment may
918 not be effective when multiple jumps are encountered. Second,
919 pushing an isovalue over a jump does not guarantee that the
920 distances on the two sides of the jump will be equal, especially
921 when the jump occurs close to the end of the value range. In ad-
922 dition, this situation is often aggravated when additional jumps
923 are encountered.

924 **Configuration selection.** As we have seen before, treating
925 jumps and spikes can both have a positive or a negative impact
926 on the achieved solution. In order to recommend an appropri-
927 ate configuration, we run experiments among all time steps of
928 different data sets and variables. Figure 14 shows the mean and
929 standard errors of the best average errors with different confi-
930 gurations using four variables from four different data sets.

931 The top row shows the results collected using all the time
932 steps. We can see that in most of the cases, one of our conver-
933 gence stabilization configurations improves the overall solution
934 or yields a solution similar to the original one. For (a), the CHI
935 variable of the combustion data set, our experiment shows that
936 treating the jumps but ignoring spikes outperforms the other
937 methods by a huge margin. This is likely due to an initial set of
938 isovalues being stuck in a local optimum and can only escape
939 it by treating jumps while allowing huge spikes. For (b), the H
940 variable of the ionization data set, we witness that a strict pol-
941 icy for spikes yields the best results. Interestingly, we can see
942 that all configurations, except for $\delta_s = 0.05, 0.1$ without jump
943 treatment, and $\delta_s = 0.2$ with jump treatment, have a very high
944 standard error. This indicates that the structure of the volume
945 differs heavily between time steps. In contrast, a loose spike
946 treatment ($\delta_s = 0.2, 0.25$) has a small standard error among all
947 configurations using the SALT variable of the climate data set,
948 as shown in (c). Similarly, in (d), the bar chart shows that too
949 strict spike treatment can have a strong negative impact on the
950 performance, using the VEL_MAG variable of the atmosphere
951 data set.

952 As there is no clear favorite among all data sets, we sam-
953 ple the first five time steps of a data set to see if we can pre-
954 dict a good configuration for the full run. These results are
955 demonstrated in the bottom row of Figure 14. In (a) and (b),
956 our method performs consistently over the time steps, showing
957 the possibility to predict the best variation from the first couple
958 of time steps. In (c), although we mispredict the best configura-

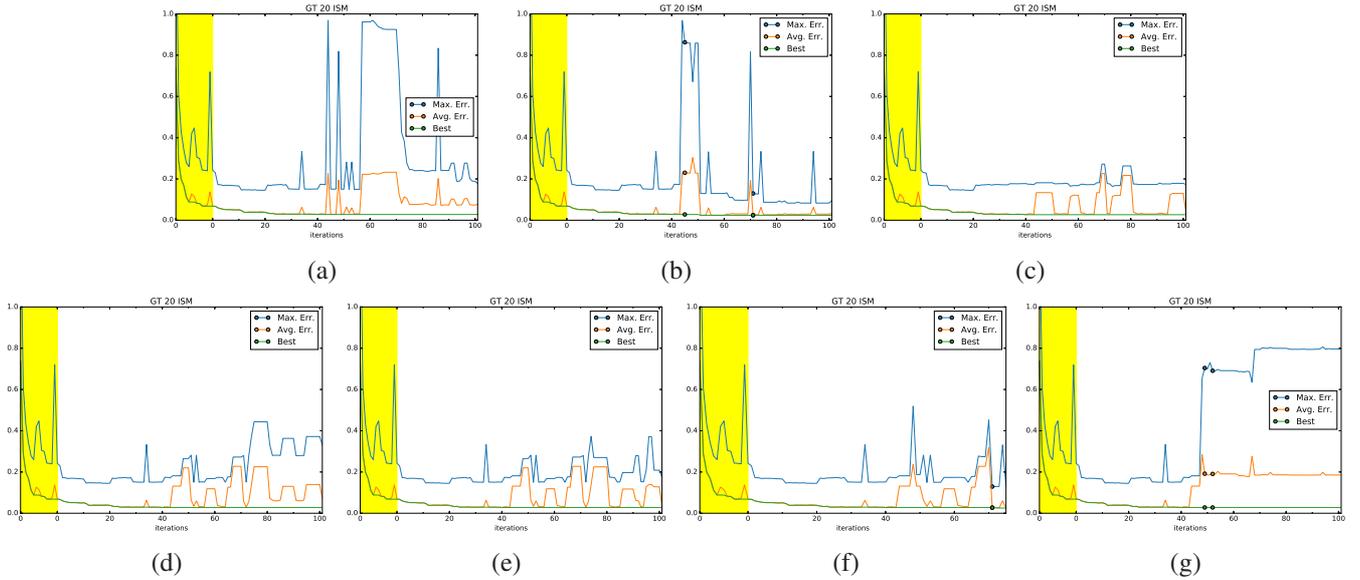


Figure 13: Error curves generated with different configurations using the GT variable of the ionization data set at time step 20. (a) shows the original approach without jump and spike treatments. (b) shows a configuration with only jump treatment. (c) to (g) show configurations with both jump and spike treatments while δ_s increases from 0.05 to 0.25 in steps of 0.05. Dots on the curves indicate the iterations when jumps are detected.

959 tion, the predicted one still shows significant improvement over
 960 the original. However, in (d), the bar chart indicates that a bad
 961 configuration is recommended based on the results using the
 962 first five time steps. Although the recommended configuration
 963 ($\delta_s = 0.15$ with jump treatment) performs similarly as the origi-
 964 nal approach, we fail to spot the best configuration ($\delta_s = 0.15$
 965 without jump treatment). In this example, the high standard errors
 966 among the first five time steps may indicate that the inter-
 967 mediate favorite is not a stable one for all the time steps. Over-
 968 all, we cannot always predict the exact configuration that yields
 969 the best result for a given data set. However, when considering
 970 the first couple of time steps, we often identify a configuration
 971 of the convergence stabilization that outperforms the original.

972 5. Conclusions

973 We have presented a solution for identifying nearly equally
 974 spaced isosurfaces for volumetric data sets. Motivated by find-
 975 ing a small set of isosurfaces to better represent the underly-
 976 ing volume data in the spatial domain, we design a two-stage
 977 approach to seeking an approximated solution that maintains
 978 a good balance between quality and performance. The result-
 979 ing surfaces are nearly equally spaced, and the user can freely
 980 choose the number of surfaces. Our study demonstrates the ef-
 981 fectiveness of the proposed approach and leads to valuable feed-
 982 back. To conclude, we summarize our key findings and major
 983 recommendations as follows.

984 First, our two-stage strategy is effective for achieving the
 985 best solution in a small number of iterations. Our experiment
 986 shows that both stages are necessary: without the estimation
 987 stage the refinement stage would require a lot more iterations to
 988 converge, and the estimation stage may never achieve a solution
 989 with a similar error as the refinement stage does.

990 Second, our approach can produce nearly equally spaced
 991 isosurfaces for most of the data sets, although some error may

992 be introduced by jump discontinuities in the distance functions.
 993 These points may divide the entire range of isovalues into mul-
 994 tiple segments and prevent the isovalues from moving between
 995 neighboring segments, resulting in unequal distances among the
 996 segments. Our convergence stabilization scheme alleviates this
 997 situation by treating spikes and jumps explicitly, although the
 998 effectiveness depends on the specific data set.

999 Finally, our approach is independent of the choice of the
 1000 distance measure. This provides great flexibility for users to
 1001 apply a suitable distance measure according to their own needs.
 1002 Our experiment performs effectively using both the ISM and
 1003 MCP distance measures with a common parameter setting of
 1004 $\langle \delta_e, \delta_\tau, \delta_r \rangle = \langle 10, 5, 20 \rangle$ for all the data sets. To ensure
 1005 smoother convergence without a strong restriction, we recom-
 1006 mend using $\delta_s = 0.2$ and ignoring jumps, as these settings either
 1007 outperform or show similar results as the original across all data
 1008 sets.

1009 Acknowledgements

1010 This research was supported in part by the U.S. National
 1011 Science Foundation through grants IIS-1456763, IIS-1455886,
 1012 and CNS-1629914, and the NVIDIA GPU Grant Program. The
 1013 authors would like to thank the anonymous reviewers for their
 1014 insightful comments.

- 1015 [1] Tenginakai S, Lee J, Machiraju R. Salient iso-surface detection with
 1016 model-independent statistical signatures. In: Proceedings of IEEE Vi-
 1017 sualization Conference. 2001, p. 231–8.
- 1018 [2] Bruckner S, Möller T. Isosurface similarity maps. Computer Graphics
 1019 Forum 2010;29(3):773–82.
- 1020 [3] Jobard B, Lefler W. Creating evenly-spaced streamlines of arbitrary den-
 1021 sity. In: Visualization in Scientific Computing 1997: Proceedings of the
 1022 Eurographics Workshop in Boulogne-sur-Mer France. 1997, p. 43–56.
- 1023 [4] Spencer B, Laramie RS, Chen G, Zhang E. Evenly spaced stream-
 1024 lines for surfaces: An image-based approach. Computer Graphics Forum
 1025 2009;28(6):1618–31.

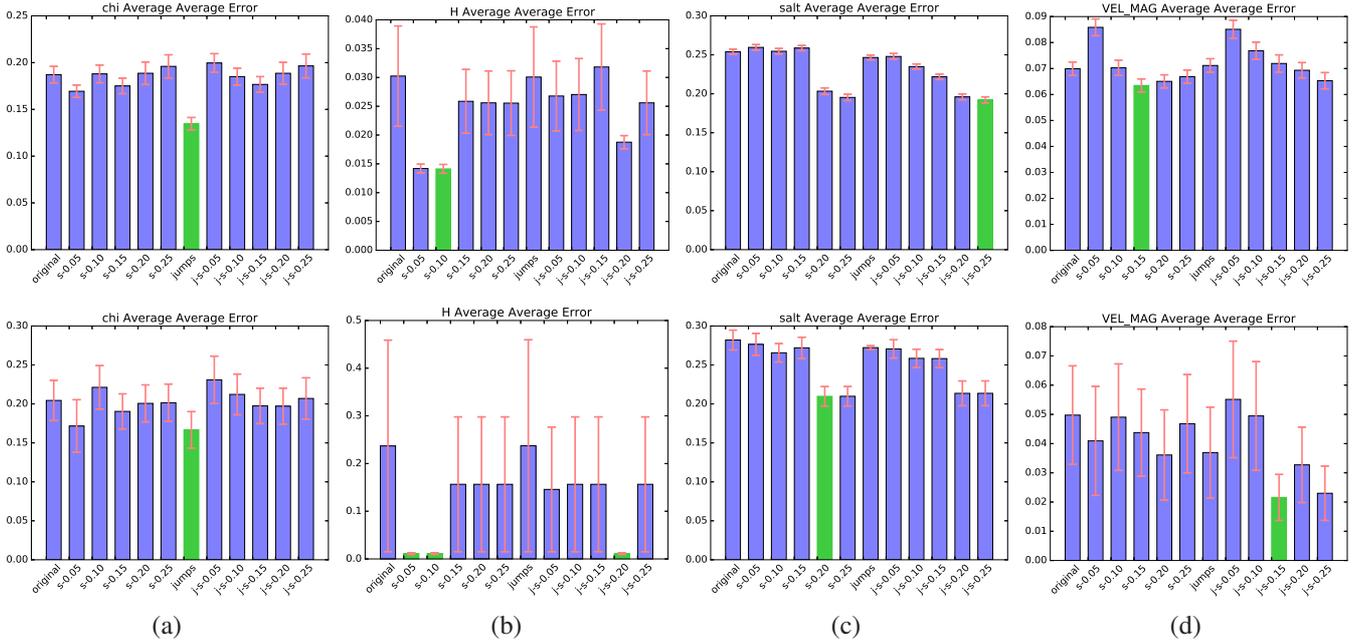


Figure 14: Comparison of the best average errors with different settings using (a) the CHI variable of the combustion data set, (b) the H variable of the ionization data set, (c) the salinity (SALT) variable of the climate data set, and (d) the velocity magnitude (VEL_MAG) variable of the atmosphere data set. The top row shows the results using all the time steps and the bottom row shows the results using only the first five time steps. In each chart, a bar represents the mean of the average errors achieved over the respective time steps of the corresponding data set (lower is better). The bars from left to right correspond to the original ($\delta_s = \infty$), spike treatment with $\delta_s = 0.05, 0.1, 0.15, 0.2, 0.25$, and jump and spike treatments with $\delta_s = \infty, 0.05, 0.1, 0.15, 0.2, 0.25$, respectively. The error bars in red indicate the standard error of the mean. The bars in green represent the best settings.

- 1026 [5] Schulze M, Martinez Esturo J, Günther T, Rössl C, Seidel HP, Weinkauff
1027 T, et al. Sets of globally optimal stream surfaces for flow visualization.
1028 Computer Graphics Forum 2014;33(3):1–10.
1029 [6] Ma B, Suter SK, Entezari A. Quality assessment of volume com-
1030 pression approaches using isovalue clustering. Computers & Graphics
1031 2017;63:18–27.
1032 [7] Moberts B, Vilanova A, van Wijk JJ. Evaluation of fiber clustering meth-
1033 ods for diffusion tensor imaging. In: Proceedings of IEEE Visualization
1034 Conference. 2005, p. 65–72.
1035 [8] Scheidegger CE, Schreiner JM, Duffy B, Carr H, Silva CT. Revisiting
1036 histograms and isosurface statistics. IEEE Transactions on Visualization
1037 and Computer Graphics 2008;14(6):1659–66.
1038 [9] Duffy B, Carr H, Möller T. Integrating isosurface statistics and histo-
1039 grams. IEEE Transactions on Visualization and Computer Graphics
1040 2013;19(2):263–77.
1041 [10] Pekar V, Wiemker R, Hempel D. Fast detection of meaningful isosurfaces
1042 for volume data visualization. In: Proceedings of IEEE Visualization
1043 Conference. 2001, p. 223–30.
1044 [11] Bajaj CL, Pascucci V, Schikore DR. The contour spectrum. In: Proceed-
1045 ings of IEEE Visualization Conference. 1997, p. 167–73.
1046 [12] Bremer PT, Pascucci V, Hamann B. Maximizing adaptivity in hierarchi-
1047 cal topological models. In: Proceedings of International Conference on
1048 Shape Modeling and Applications. 2005, p. 298–307.
1049 [13] Bremer PT, Weber G, Tierny J, Pascucci V, Day M, Bell J. Interactive
1050 exploration and analysis of large-scale simulations using topology-based
1051 data segmentation. IEEE Transactions on Visualization and Computer
1052 Graphics 2011;17(9):1307–24.
1053 [14] Carr H, Snoeyink J, van de Panne M. Flexible isosurfaces: Simplifying
1054 and displaying scalar topology using the contour tree. Computational
1055 Geometry 2010;43(1):42–58.
1056 [15] Correa CD, Lindström P, Bremer PT. Topological spines: A structure-
1057 preserving visual representation of scalar fields. IEEE Transactions on
1058 Visualization and Computer Graphics 2011;17(12):1842–51.
1059 [16] Haidacher M, Bruckner S, Gröller E. Volume analysis using multimodal
1060 surface similarity. IEEE Transactions on Visualization and Computer
1061 Graphics 2011;17(12):1969–78.
1062 [17] Wei TH, Lee TY, Shen HW. Evaluating isosurfaces with level-set-based
1063 information maps. Computer Graphics Forum 2013;32(3):1–10.
1064 [18] Krishnamurthy A, McMains S, Haller K. GPU-accelerated minimum
1065 distance and clearance queries. IEEE Transactions on Visualization and
1066 Computer Graphics 2011;17(6):729–42.
1067 [19] Larsen E, Gottschalk S, Lin MC, Manocha D. Fast distance queries with
1068 rectangular swept sphere volumes. In: Proceedings of IEEE International
1069 Conference on Robotics and Automation. 2000, p. 1–8.
1070 [20] Liu F, Kim YJ. Exact and adaptive signed distance fields computation for
1071 rigid and deformable models on GPUs. IEEE Transactions on Visualiza-
1072 tion and Computer Graphics 2014;20(5):714–25.
1073 [21] Seitzer J. Parallel computation of the Euclidean distance transform on a
1074 three-dimensional image array. IEEE Transactions on Parallel and Dis-
1075 tributed Systems 2003;14(3):203–12.
1076 [22] Yu H, Xie J, Ma KL, Kolla H, Chen JH. Scalable parallel distance field
1077 construction for large-scale applications. IEEE Transactions on Visual-
1078 ization and Computer Graphics 2015;21(10):1187–200.
1079 [23] Karras T. Maximizing parallelism in the construction of BVHs, octrees,
1080 and k-d trees. In: Proceedings of ACM SIGGRAPH/Eurographics Con-
1081 ference on High-Performance Graphics. 2012, p. 33–7.
1082 [24] Imre M, Tao J, Wang C. Efficient GPU-accelerated computation of iso-
1083 surface similarity maps. In: Proceedings of IEEE Pacific Visualization
1084 Symposium. 2017, p. 180–4.
1085 [25] Lloyd S. Least squares quantization in PCM. IEEE Transactions on In-
1086 formation Theory 1982;28(2):129–37.