

Scalar Field Visualization via Extraction of Symmetric Structures

Talha Bin Masood · Dilip Mathew Thomas · Vijay Natarajan

Abstract Identifying symmetry in scalar fields is a recent area of research in scientific visualization and computer graphics communities. Symmetry detection techniques based on abstract representations of the scalar field use only limited geometric information in their analysis. Hence they may not be suited for applications that study the geometric properties of the regions in the domain. On the other hand, methods that accumulate local evidence of symmetry through a voting procedure have been successfully used for detecting geometric symmetry in shapes. We extend such a technique to scalar fields and use it to detect geometrically symmetric regions in synthetic as well as real-world datasets. Identifying symmetry in the scalar field can significantly improve visualization and interactive exploration of the data. We demonstrate different applications of the symmetry detection method to scientific visualization: query-based exploration of scalar fields, linked selection in symmetric regions for interactive visualization, and classification of geometrically symmetric regions and its application to anomaly detection.

Keywords Scalar field visualization · Symmetry detection · Query based exploration.

1 Introduction

Scalar field data have been widely used to represent physical quantities measured over a domain of interest and the

underlying properties of the physical phenomena are studied by examining the distribution of the scalar values. Often these distributions contain repeating or symmetric patterns and they provide significant insights that aid in the study of the physical phenomena. For example, in X-ray crystallography, the symmetric patterns in the intensity of the diffracted rays play an important role in determining the structure of the crystal. Automatic detection and visualization of symmetry in the scalar field can considerably simplify the challenges faced by the domain scientists in exploring the data and hence there is a lot of research interest to develop such techniques.

One of the main challenges in extracting symmetry information from scalar fields is its computational cost since scalar field datasets are typically large in size. To reduce the computational cost, abstract representations of the scalar field that are much smaller in size like the contour tree [2] and constellation of crease-line features [11] have been used in the past for symmetry detection [12, 16]. Instead of searching for symmetry in the entire dataset, these methods assume that the symmetry in the data will also be reflected in the abstract representation and search for symmetric structures in the abstract representation. Though the reduction in the size of the search space significantly improves the computational efficiency, symmetric patterns in the abstraction may not correspond to the symmetry in the data and vice versa. We describe these methods below and believe that these abstractions are often not ideal for encoding the symmetry in the underlying data.

The contour tree is an abstract representation that captures the topology of the level sets of the scalar field. Thomas and Natarajan [16] propose a method to determine the similarity between two subtrees in the contour tree through a similarity measure based on the extent of the overlap between the subtrees. The similarity measure is then used to classify the subtrees of the contour tree into different groups. The regions of the domain corresponding to the

T. B. Masood and D. M. Thomas
Department of Computer Science and Automation,
Indian Institute of Science, Bangalore.

V. Natarajan
Department of Computer Science and Automation,
Supercomputer Education and Research Centre,
Indian Institute of Science, Bangalore.

E-mail: {tbmasood,dilip,vijayn}@csa.iisc.ernet.in

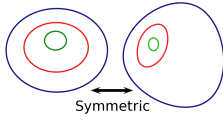


Fig. 1 Non-intuitive symmetry detected using contour tree approach.

subtrees that belong to the same group are reported to be symmetric. However, since the contour tree is a topological data structure, this method does not ensure that the geometry of the detected regions indeed exhibit symmetry. For example, in Figure 1, the region on the left and the right would be identified as symmetric since the topology of the level sets is the same even though the geometry of the two regions are very dissimilar. Kerber et al. [12] identify crease-lines in the scalar field and determine the junction points at which the crease-lines intersect. The geometric transformation that maps the network of lines in one junction to another is then computed. Each such transformation is applied to the data and the symmetric regions are identified by measuring the deviation in the scalar values between the original and the transformed regions. However, typical scalar field datasets contain far more complex features than just crease-lines. Hence, though the abstraction used is based on geometry, it severely limits the symmetric regions that can be detected by the technique.

A wide range of techniques have been studied for robustly estimating symmetry in shapes as described in the recent survey by Mitra et al. [15]. Since these methods use geometric properties of shapes to identify symmetry, extending them to scalar fields, though often not trivial, has the obvious advantage of being able to detect symmetry in scalar fields in a geometry-aware manner. Kazhdan et al. [10] detect reflective symmetry by using a voxel based representation for shapes and measuring the degree of reflectivity for each candidate plane. Hong et al. [9] extend this idea to scalar fields and measure the difference in the scalar value of a point and its reflection to determine the symmetry distance of a plane. In addition to being computationally expensive, the main drawback of this method is that it is restricted to detecting only global reflective symmetry whereas our method can detect partial rotational and translational symmetries also.

Mitra et al. [14] propose a two-step solution for identifying partial and approximate symmetry. In the first step, points with similar local shape signature are paired together and each such pair votes for a symmetry transformation in a transformation space. Symmetry detection problem can then be reduced to finding clusters in this high dimensional space where each cluster corresponds to aggregation of votes for a particular symmetry transformation. Hence, in the second step, such clusters are identified and the corresponding symmetry transformation is spatially verified and the symmetric patches are extracted. This method is computationally efficient since statistical evidence of symmetry is first ac-

cumulated before performing the costly operation of spatial verification and can also detect partial symmetries in the data. Moreover, the clustering mechanism ensures robustness in the presence of noise. For these reasons, we adopt this approach to detect partial symmetries in scalar field in a geometry-aware manner. Extending this method to scalar field involves significant challenges and are described in Section 3.

Symmetry plays an important role in enhancing visualization and post-processing of the scalar field data. In volume visualization, knowledge of the repeating patterns in the domain helps in designing symmetry-aware transfer functions that classifies regions of the domain into different groups based on their similarity [16]. Similarly, isosurface components can be classified into different symmetric groups and selectively visualized for better interactive exploration of isosurfaces [16]. Symmetry detection also helps in selection of meaningful cross-section planes for slicing volume data, better visualization of features by applying different rendering techniques that show complementary information in the symmetric regions and selection of view directions that removes redundancies in the data [9]. In section 4, we demonstrate different methods that make use of symmetry information to enhance scientific data visualization and exploration.

Contributions. We make the following contributions towards a symmetry-aware approach to scalar field visualization:

- A generic voting-based symmetry identification pipeline for 2D and 3D scalar fields.
- Local function descriptors for different domains viz. plane, 3D volume, and 2-manifolds, that are used to detect pairs of points or pairs of regions that have a similar scalar field distribution.
- A symmetry-aware method for query-based exploration of scalar fields.
- A method for linked selection in symmetric regions and its application to interactive visualization of scalar fields.
- A method for classification of geometrically symmetric regions using scalar field symmetry and its application to anomaly detection.

2 Background and Definitions

A *scalar field*, s , is a scalar function defined on a manifold, $\mathbb{M}$ i.e. $s : \mathbb{M} \rightarrow \mathbb{R}$. In this paper, we consider scalar fields defined on 2-manifolds and $\mathbb{R}^3$. In practice, the scalar field is available as a discrete sample at vertices of a simplicial mesh or structured grid that represents the domain. Each vertex in a mesh is assigned a scalar value. Scalar values at other points in the mesh are obtained via interpolation.

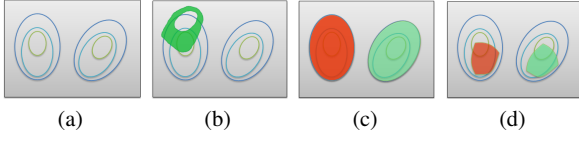


Fig. 2 (a) Three level sets of a scalar field s are shown in different colors. (b) An arbitrary region in s . (c) A symmetric region pair. (d) A symmetric pair which is not maximal.

Level set L for a given value v is the preimage of s i.e. $L(v) = s^{-1}(v)$. Level sets of 3D scalar fields are called *iso-surfaces*, while the level sets of 2D scalar fields are called *isocontours*. The gradient of a scalar field s is a vector field that points in the direction of the greatest rate of increase of the scalar field, and whose magnitude is the greatest rate of change.

A connected subset $r \subseteq \mathbb{M}$ of the domain $\mathbb{M}$ of a scalar field s is called a *region* of s . Two regions r_1 and r_2 are called as symmetric if there exists a valid transformation T such that $T(r_1) = r_2$ and $s(p) = s(T(p))$ for all $p \in r_1$. This definition captures exact symmetry which is rarely observed in real datasets or in synthetic datasets because of discretization errors. So, we define a notion of approximate symmetry where two regions, r_1 and r_2 , are symmetric if $T(r_1) = r_2$ and for all points $p \in r_1$, $s(p) = s(T(p)) + \epsilon$, for a small real value ϵ . We restrict the transformations to rigid body transformations of the domain.

A pair of symmetric regions is called a *symmetric region pair* if it satisfies the condition of maximality i.e. no more points of the domain can be added to the regions under the same transformation. More formally, a pair (r_1, r_2) is called a symmetric region pair under transformation T if there does not exist any region $r'_1 \supset r_1$ such that the pair $(r'_1, T(r'_1))$ is also symmetric under T . The notion of regions and maximality is illustrated in Figure 2. We define *significance* of the symmetric region pair as the volume of that pair.

Now, the problem of identifying symmetry is stated as that of identifying all symmetric region pairs in the scalar field with significance greater than a specified threshold.

3 Symmetry Detection Pipeline

Voting-based techniques have been used for identifying symmetry in geometric shapes earlier [14, 15]. Mitra et al. [14] propose a technique where evidence for symmetries is accumulated by comparing the local signatures of sufficient number of samples. The evidence accumulation is accomplished by identifying the transformations between pairs of regions that are locally similar and each such pair votes for a transformation in the space of all transformations. If many pairs vote for the same transformation then the number of votes for that transformation becomes high. Significant symmetries appear as dense cluster of votes in

the transformation space. This evidence is verified by reverting back to the spatial domain and the resulting regions are reported to be symmetric.

We extend the voting-based technique for detecting symmetry in scalar fields defined on the plane, 3D volumes and 2-manifolds. This extension to scalar fields poses many challenges. A new notion of local signatures for the scalar field is required which helps capture invariance under transformations. Principal or key directions need to be identified to determine the geometric transformations that best aligns local regions of the scalar field.

As a motivation for the solution, let us observe the effect of rigid transformation on a 2D scalar field. It is clear that after geometric transformations of a region, quantities like scalar values, gradient magnitude and curvature of the contour remain unchanged. The direction of gradient and tangent to the contour are transformed according to the specified transformation. Since symmetry detection involves finding regions which are invariant under transformations, the observation above leads us to search for point pairs in the scalar field that have the same scalar value, gradient magnitude and contour curvature. Each such pair thus provides local evidence for the presence of a symmetric region. For each pair, given the gradient direction and position of the points, we determine the transformation. For large symmetric regions, a large number of pairs vote for the same transformation and will appear as dense clusters in the transformation space. So, the problem of finding symmetric regions in a scalar field is reduced to finding pairs that can vote for transformations and clustering in transformation space to obtain symmetric regions. The reader is referred to Figure 3 for an illustrative example. We now describe our proposed pipeline for identifying symmetric regions in a scalar field.

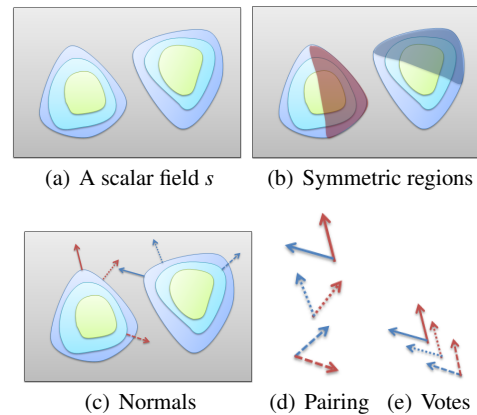


Fig. 3 (a) Three level sets of a scalar field are shown in different colors. (b) Symmetric regions in this scalar field. The shaded regions are geometrically symmetric. (c) Gradients of the ideal pairs in the symmetric regions. (d) These gradient vectors are paired up. (e) Angles between the paired gradient vectors are equal. So, these ideal pairs vote for the same transformation.

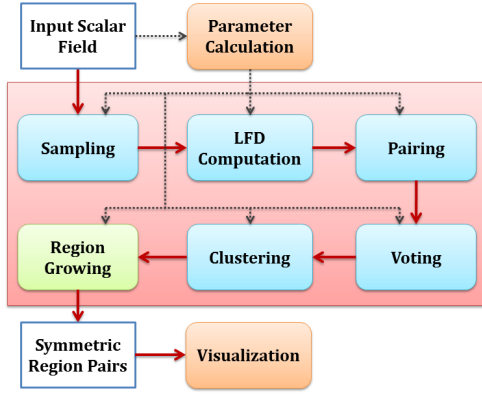


Fig. 4 Symmetry Identification Pipeline.

3.1 Overview

We employ a two stage approach for identifying symmetry in scalar fields. The stages of the pipeline are:

1. Evidence accumulation stage: Accumulate evidence for various symmetries.
2. Validation stage: Verify the validity of symmetries obtained in the previous stage by region growing.

The pipeline is illustrated in Figure 4. The blue blocks in the figure are substages of evidence accumulation stage. We discuss these stages in detail below. Figure 5 illustrates the working of the pipeline on a simple 2D scalar field. First, the input is sampled. Next, some of these samples are paired up on the basis of local properties. These pairs are shown in 5(c). The pairs then vote for transformations in the space shown in 5(d). Clusters in this space provide evidence for the symmetries, which is translation for this example. The region growing stage then generates the symmetric pair of regions as shown in Figure 5(e).

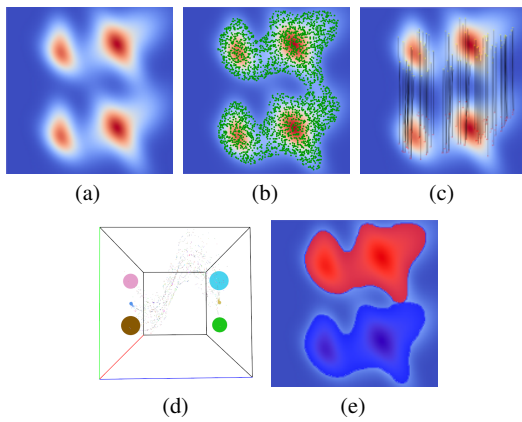


Fig. 5 Application of symmetry identification pipeline on an example 2D scalar field. (a) The input scalar field. (b) Samples. (c) Some of the pairs selected for voting. (d) Clusters in the transformation space. (e) An extracted symmetric region pair.

3.2 Sampling

We sample the input domain because the number of points in the mesh can be very large. It should be noted that as we increase the sampling rate, new symmetries may be found. Further increasing the sampling rate will not result in new symmetries, the increased sampling will merely provide more evidence for previously found symmetries. So, sampling strategy along with optimum sampling rate plays an important role in increasing the efficiency of this pipeline. We sample the mesh uniformly at the regions where gradient is greater than a specified threshold. This ensures that flat regions are ignored.

3.3 Local function descriptor computation

Local function descriptor. For each point sample p we compute the *local function descriptor* (LFD). The LFD captures the local function distribution around a point p on the domain. The LFD is used by the pairing and voting stages of the pipeline. The LFD consists of an invariant component and an alignment component. The invariant component consists of properties that remain the same under rigid transformations e.g. scalar value, gradient magnitude, and curvature of the level set passing through the point. The alignment component consists of vectors that provide a consistent coordinate reference frame at each point. The gradient vector together with the principal curvature directions of the local level set act as a local frame. We employ different LFDs for different domains as listed in Table 1.

If the LFDs of two points have similar invariant components, then it is highly likely that the domain geometry as well as the local distribution of the scalar field around the points are the same. So, the local regions containing the points are symmetric and these regions potentially belong to larger symmetric regions, which would be detected later. The scalar value and gradient magnitude are invariant under transformation. Also, transformation of gradient vectors in two regions is same as the transformation of the corresponding regions. Therefore, gradient vector is a part of the alignment component of LFD. Similar to the gradient, the level sets in the neighborhood of a point are also transformed with properties like curvature remaining unchanged. These local properties are sufficient for domains like $\mathbb{R}^2$ and $\mathbb{R}^3$. However, for general 2-manifolds we need to ensure that in addition to the distribution of the scalar field, the domain geometry should also be similar for the two points to be paired up. So, we introduce the curvature of the domain at the points as an invariant component and the normal to the surface as one of the alignment components.

LFD computation. The gradient vector at a point p is approximated by computing the difference in function values between its neighbors in the mesh. For 2D scalar fields, the

Table 1 Local Function Descriptors of a point in the scalar field domain

Domain	Local Function Descriptor of a point p	
	Invariant Component	Alignment Component
$\mathbb{R}^2$	1. Scalar value, $s(p)$ 2. Magnitude of the gradient, $ \nabla s(p) $ 3. Curvature of contour $L(s(p))$ at p	1. Gradient vector, $\nabla s(p)$
2D Manifold	1. Scalar value, $s(p)$ 2. Magnitude of the gradient, $ \nabla s(p) $ 3. Curvature of contour $L(s(p))$ at p 4. Curvature of domain i.e. k_{max} and k_{min} of the surface at p	1. Gradient vector, $\nabla s(p)$ 2. Surface normal, n 3. Cross product, $\nabla s(p) \times n$
$\mathbb{R}^3$	1. Scalar value, $s(p)$ 2. Magnitude of the gradient, $ \nabla s(p) $ 3. Curvature of isosurface $L(s(p))$ at p i.e. k_{max} and k_{min}	1. Gradient vector, $\nabla s(p)$ 2. Principal curvature directions of isosurface $L(s(p))$ at p i.e. PC_{max} and PC_{min}

local isocontour at p restricted to the 1-ring neighborhood is extracted. The curvature is approximated as the inverse of the radius of the circle that best fits the contour. This is illustrated in Figure 6. A better curvature approximation can be obtained by considering 2-ring (k -ring) neighbors of the point and fitting a spline. For 3D scalar fields, a local isosurface is extracted from the tetrahedral mesh. If the domain is represented using a structured grid, then voxels incident on p are subdivided into tetrahedra. The local isosurface may also be extracted directly using the marching cubes algorithm [13]. Once the local isosurface is extracted, the principal curvatures and directions at p are computed by estimating the curvature tensor at p [1, 4].

3.4 Pairing

In the pairing stage, we find the point pairs that provide evidence for symmetry in the scalar field. As a first step, we prune the sample set P . The points where the minimum curvature (k_{min}) is equal to the maximum curvature (k_{max}) are called umbilic points. At such points curvature is equal in all directions, thus we can not find a consistent coordinate frame for alignment for this point. So, we remove such points from P to obtain P' .

All pairs of points in P' are compared. If the invariant component of the pairs are equal or within user defined threshold, then they are added to the voter pairs set V . The

pairing is speeded up by building a kd-tree on the invariant component of LFD of points. For each point, we determine its pairs by a range query on the kd-tree. The range query is built for each point using the tolerance threshold allowed for each invariant component of LFD. The output of the pairing stage is the set of point pairs V .

3.5 Voting

In this stage each point pair $(p_i, p_j) \in V$ votes for a transformation. A transformation in the plane is represented as a point in $\mathbb{R}^3$, $T_{ij} = (t_x, t_y, r)$, where (t_x, t_y) is the 2D translation vector and r is the rotation angle. For the case of general 2-manifolds and 3D Euclidean space, a transformation is represented as a point in $\mathbb{R}^6$, $T_{ij} = (t_x, t_y, t_z, r_x, r_y, r_z)$ where (t_x, t_y, t_z) is the 3D translation vector and (r_x, r_y, r_z) are the Euler angles for rotation.

The transformation is computed from the alignment component of the LFDs of p_i and p_j . For determining the transformation vector, first the local coordinate frames of the points p_i and p_j are aligned. Let R be the rotation matrix corresponding to the alignment. The Euler angles (r_x, r_y, r_z) are determined from this rotation matrix R . The translation vector (t_x, t_y, t_z) is computed as $p_j - Rp_i$. The output of the voting stage is a set of transformations $\mathbb{T}$.

3.6 Clustering

To determine significant symmetries in the scalar field, clustering is performed in the transformation space. We have various options for performing clustering like single linkage clustering, DBSCAN [7] and mean shift clustering [5]. We believe mean shift clustering would give the best clusters for our purpose, as it is a density gradient based method where each point in the space converges to the basin of attraction. However, during experimentation, we found that even single linkage clustering which is much simpler and faster as

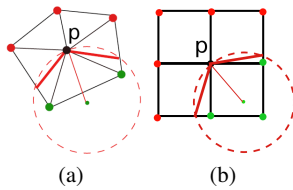


Fig. 6 The contour passing through the point p is shown in bold red. The curvature is computed by taking the inverse of the radius of circle fitted to the contour, shown in red dashed line.

compared to the other two clustering algorithms, gives satisfactory results.

The significance of a cluster is determined by the number of points assigned to that cluster. A cluster in the transformation space is large only if many point pairs vote for the same transformation, thus indicating a large symmetric region. We rank the clusters in decreasing order of their significance, and compute the centroid of the cluster as the transformation representative of the cluster. The output of the clustering stage is a ranked list of transformations.

3.7 Validation stage

Transformations obtained from the clustering stage may not necessarily correspond to a single significant symmetry. It may happen that pairs from different non-significant regions vote for the same transformation, thus resulting in a large cluster. We prune such transformations during the validation step, and also re-rank the transformations.

During validation stage, region growing is done to determine the actual regions that voted for a particular transformation. Region is grown for every significant cluster. This process is initiated by picking a random pair from the cluster and growing the regions in breadth first fashion. When no more points can be added to the region without violating the approximate symmetry threshold, we stop the region growing process for that pair. If the grown regions covers all the points in the cluster, then region growing for the cluster is stopped and we move on to the next cluster. However, if some points are not covered by region growing of initial pair, which will happen when disconnected regions have voted for the same transformation, we repeat the region growing process for the next uncovered pair in the cluster. This process continues until there are no uncovered pairs left in the cluster.

It should be noted that the region growing process ensures that we get connected and maximal regions as required. The symmetric region pairs (r_i, r_j) which don't satisfy the significance threshold are ignored while others are added to the output along with the associated transformation under which the pair is symmetric.

4 Application of Symmetry in Visualization

Symmetry information can be exploited for generating better visualizations of scalar fields.

The simplest technique of using symmetry in visualization is by showing all the symmetric pairs of regions individually in a sequence of images. For example, for a slice from a 3D probability distribution of electrons in a hydrogen molecule as shown in Figure 7(f), two pairs of regions symmetric under rotation by 180° are detected. These pairs

are shown separately in Figures 7(g) and 7(h). Similarly, Figure 7(j) shows a symmetric pair of regions detected in the velocity field of a von Karman vortex street flow simulation. Figure 7(b) shows the temperature distribution on the surface of a conductor element modeled and simulated using COMSOL software. The symmetric patches detected in this dataset are shown in Figure 7(c). Figure 7(d) shows the pressure field of a Taylor-Green vortex flow simulation. This dataset has high degree of symmetry. One of the many symmetric regions is visualized in Figure 7(e).

We now discuss a few applications of symmetry in exploration and visualization of scalar fields.

4.1 Query-based exploration

Suppose the user identifies an interesting region in the scalar field and she is interested in repeating occurrences of this region. To support this, selection of regions using hierarchical Morse decomposition of the scalar fields is done. One of the Morse cells is selected as the query region. Morse segmentation is chosen because it is well known technique of segmentation and the regions correspond to important features in the scalar fields [3, 6, 8]. It should be noted that any segmentation of the domain can be used since our symmetry identification method is not affected by the choice of the segmentation technique.

The query region is searched in the scalar field using our symmetry identification pipeline. A modified version of the pipeline is used to accomplish this efficiently. In addition to sampling the domain as before, the query region is also sampled. Then LFDs for all samples are computed but in the pairing stage points are paired such that one point is from query region while the other point is from the domain. This greatly reduces the number of pairs which vote and thus all the regions symmetric to the given region can be identified efficiently. This approach can be extended to build a region retrieval system wherein a query region is provided which is searched in the database of scalar fields. The system can output all the scalar fields containing the input region ordered by the frequency of occurrence and symmetric exactness of regions similar to the query.

Query-based exploration can be applied on scalar fields defined on any domain, however, for the demonstration of this technique we will focus on scalar fields defined on general 2-manifolds and 3D volumes. In Figure 8, query-based exploration of scalar fields defined on molecular surfaces and channels is shown. Skin surfaces are used for generating the molecular surfaces and channels, while APBS software is used for generating the electrostatic potential fields. Figure 8(a) shows trans-membrane protein called Mechanosensitive channel of small conductance (PDB-id:2OAU). One of the Morse cell is selected as query. The six channels symmetric to the query are successfully detected as shown in

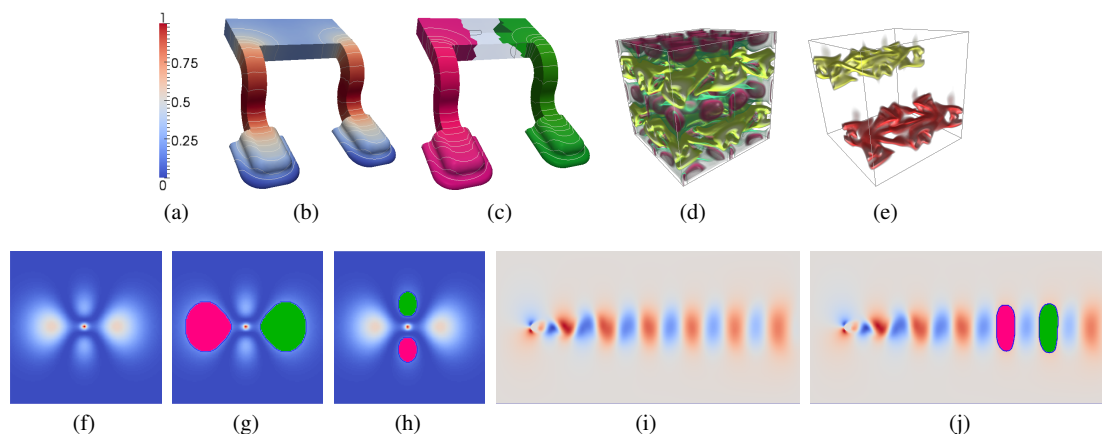


Fig. 7 Visualization of symmetry. (a) A standard diverging color map called *Cool to warm*. This color map is the default color map used in this paper. (b) COMSOL Conductor dataset. (c) A pair of symmetric regions visualized by using different colors for the regions. (d) A Taylor-Green vortex flow simulation dataset. (e) A pair of symmetric regions identified in this dataset. (f) A slice from a 3D probability distribution of electrons in a hydrogen molecule. (g) A pair of symmetric regions. (h) Another pair of symmetric regions detected in the slice. (i) von Karman vortex street flow simulation. (j) One of the identified pair of symmetric regions.

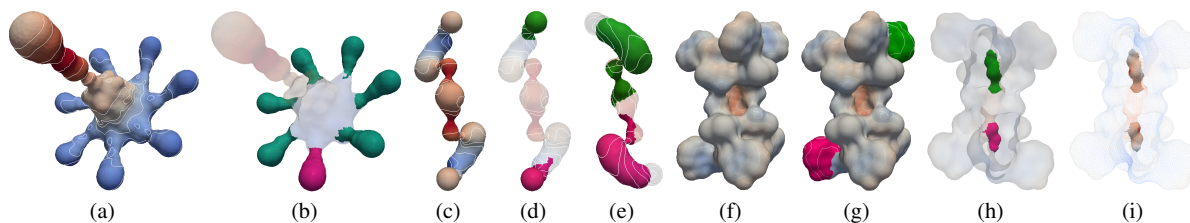


Fig. 8 Query based exploration on scalar fields defined on 2-manifolds. In these figures the query regions are shown in pink while the results are shown in green. (a) Skin surface of the channel network in a membrane protein mapped with the electrostatic field. (b) Query region and result. Six channels are symmetric to the query. (c) A channel in Gramicidin mapped with electrostatic potential. (d) A Morse cell as query and the symmetric region. (e) A similar result for another Morse cell as query. (f) Molecular surface of Gramicidin mapped with the electrostatic potential field. (g) A query region along with the result. (h) A scenario where the query and result regions are completely occluded by the molecular surface. (i) To visualize the occluded regions, the context is shown in translucent wire frame.

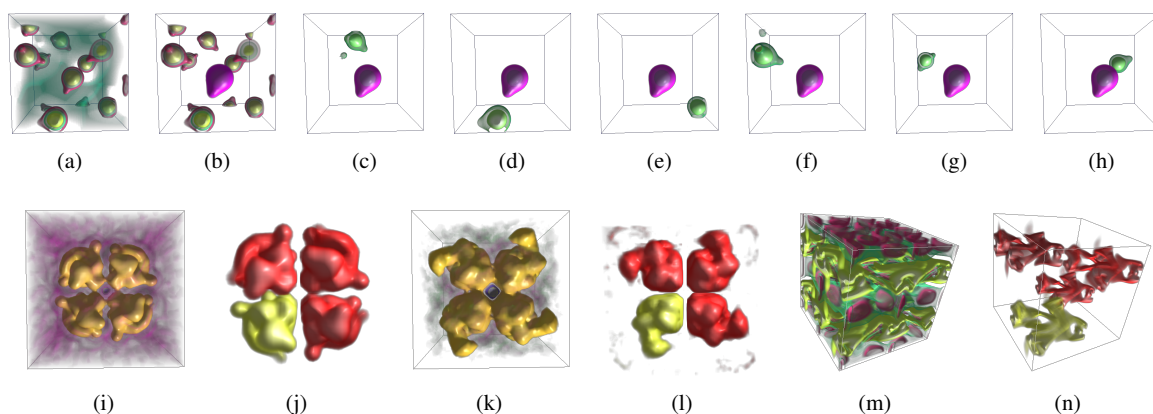


Fig. 9 Query based exploration on 3D scalar fields. (a) A scalar field representing ab-initio molecular dynamics simulation on water molecules. The water molecules are distributed in arbitrary orientations across the domain. (b) We selected one water molecule as a query region (pink) using Morse decomposition. (c) – (h) The symmetry detection pipeline identifies all the other water molecules in the domain. Each figure shows the query molecule in pink while the symmetric region to it is shown in green. (i) A Cryo-EM dataset (EMD-id:1319) with four fold symmetry. (j) We selected a query region which is highlighted in yellow. The symmetry identification pipeline successfully identifies the three symmetric regions which are shown in red. (k) and (l) A similar result for another Cryo-EM dataset (EMD-id:1654) with four fold symmetry. (m) Pressure field of a Taylor-Green vortex flow simulation. (n) The query which is one octant of the dataset is shown in yellow while the three symmetric octants are shown in red.

Figure 8(b). Figure 8(c) to Figure 8(e) show similar results for primary channel through Gramicidin (PDB-id:1GRM). Two queries along with their results on the surface of the channel are shown. We also show results for molecular skin surface of Gramicidin. Figures 8(f) to 8(i) show results for two different queries. As shown in Figure 8(g), the first query detects only one region symmetric to it, even though that residue repeats eight times in the molecule. We correctly get only one of these regions because the electrostatic potential is different at other patches due to different neighborhoods. The second query as shown in Figure 8(h) is an example where the query and the result are occluded by the molecular surface. To handle such cases we use translucent wire frame of the complete dataset to give context to the symmetric regions.

In Figure 9, results for query-based exploration of 3D scalar fields are shown. The first dataset which is considered is ab-initio molecular dynamics simulation of water in which water molecules are distributed in arbitrary orientations in the domain. This is shown in Figure 9(a). Using Morse decomposition, one of the water molecule is selected as the query. The selected region is shown in pink in Figure 9(b). With the selected molecule as a query, all the other water molecules in the field are found as shown in Figures 9(c) to 9(h). In each of these figures, the query region is shown as pink while the other region in symmetric pair is shown in green. Two Cryo-EM datasets with four-fold rotational symmetry are shown in Figures 9(i) and 9(k). These datasets represent measured quantities and are therefore noisy. We successfully detect symmetric regions even in the presence of noise, as shown in Figures 9(j) and 9(l). Lastly, we show a result for a Taylor-Green vortex flow simulation. The dataset is shown in Figure 9(m). Here, the query is chosen as one of the quadrant of the dataset. Three other quadrants which are symmetric to the query region are obtained as shown in Figure 9(n).

4.2 Symmetry-aware linked selection and interaction

Using symmetry identification, we identify a group of symmetric regions. Since we have information of the transformation between any two symmetric regions we know the point correspondences. We use the correspondences to allow linked selection of subregions in these symmetric regions. The user can choose a region from the group of symmetric regions and explore it further. If a subregion in this region is selected, the same selection is automatically reflected in all the symmetric regions. The modes of selection can be varied e.g. thresholding based on function value, clipping by a plane, etc. The user can assign a transfer function to a subregion which will automatically be used for the remaining symmetric subregions. Refer to Figure 10 for an example of this technique for 2D scalar fields.

4.3 Classification of symmetric patches

For scalar fields defined on 2-manifolds, we have symmetry at two levels. Firstly, there is symmetry in the domain. Secondly, we have symmetry when we consider the scalar field. Clearly, two regions identified as symmetric while considering the scalar field have symmetric domains as well. However, the reverse is not true. So, here we first identify a set of regions whose domains are geometrically symmetric, then we classify this set by considering the scalar field based symmetry.

This technique is demonstrated in Figure 11. We show our results for heat sink dataset. This dataset is generated using COMSOL software. The heat sink has ten geometrically symmetric plates as shown in Figure 11(c). However, all the plates are not symmetric if we consider the temperature distribution. We find four pairs of symmetric plates as shown in Figures 11(e) to 11(h). So, the original group of symmetric plates are further classified into different symmetric groups based on scalar field distribution. The new grouping obtained is shown by giving distinct color to different groups symmetric patches. This color coded classification is shown in Figure 11(d).

4.4 Anomaly detection

The previous technique can also be used for anomaly detection. Suppose the domain has symmetric structures. We know beforehand what the ideal scalar field distribution should be for one such group of structures. We can use our scalar field symmetry identification method to detect the structures which do not have ideal scalar field distribution. These structures may point to some anomaly e.g. consider the stress distribution on the columns of a bridge. Ideally each column should have similar stress distribution, but if our method identifies a column which has significantly different stress distribution then it may point to some structural defect in the bridge.

We demonstrate this technique in Figure 12. We obtained a 3D model of crab and generated a new mesh by chopping off three of its legs from one side as shown in Figure 12(a). Now, the average geodesic distance for the modified model is computed. Because of asymmetry in the new model, the distribution of average geodesic field becomes different for one of the legs while it is comparable for the remaining legs. If we identify symmetric regions in the model without taking into account the scalar field, then all the legs are identified as symmetric as shown in Figure 12(c). However, if we execute our scalar field symmetry pipeline then one of the legs will not be identified as symmetric to other members of the group as shown in Figure 12(e). We report that region as anomalous and in such cases it may point to defect.

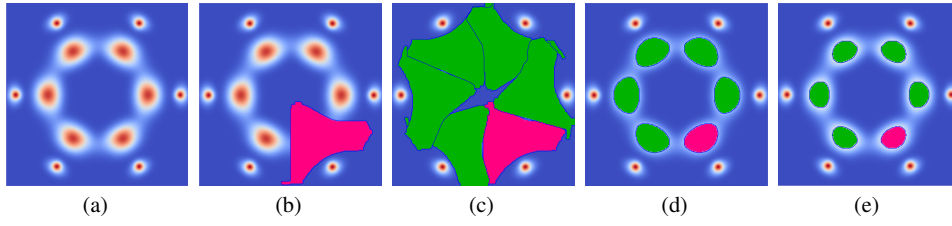


Fig. 10 Symmetry aware selection and editing. (a) A synthetic 2D scalar field with six fold symmetry. (b) A Morse cell selected as a query region. (c) Symmetry detection identifies all the regions symmetric to the query. The region chosen for further exploration is shown in pink, while symmetric regions are shown in green. (d) All the symmetric regions are linked to the chosen region. Any selection made in the chosen region is reflected in regions symmetric to it. Here, a subregion is selected in chosen region based on function value thresholding. Corresponding set is automatically selected in other regions. (e) The subregion can be interactively modified.

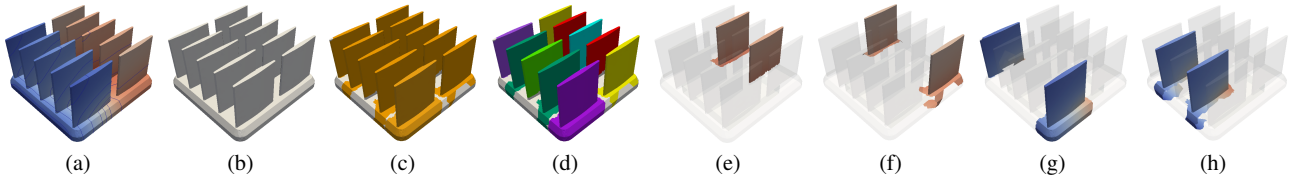


Fig. 11 Classification of domain symmetries based on symmetry in scalar field. (a) The heat sink dataset. (b) The domain of heat sink dataset. (c) The geometry based symmetry extraction on the domain identifies all the plates as symmetric. (d) The symmetry in scalar field distribution on these plates is used to classify the plates in different symmetric group. Plates which are symmetric to each other with scalar field are given same color. (e)–(h) The plates which are symmetric are shown with their scalar field distributions in the context of the domain.

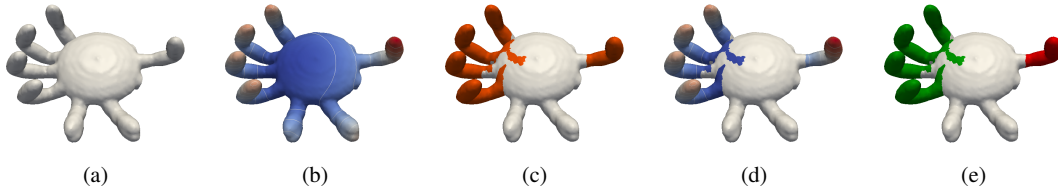


Fig. 12 Use of scalar field symmetry for anomaly detection. (a) A 3D model of crab with three of its legs missing from one side. (b) Average geodesic distance field. (c) Geometry based symmetry extraction on the domain identifies all the five legs as symmetric. (d) Scalar field distribution is different on one leg as compared to others. (e) The anomalous leg is shown as red while the legs shown in green are symmetric in terms of scalar field distribution as well as domain geometry.

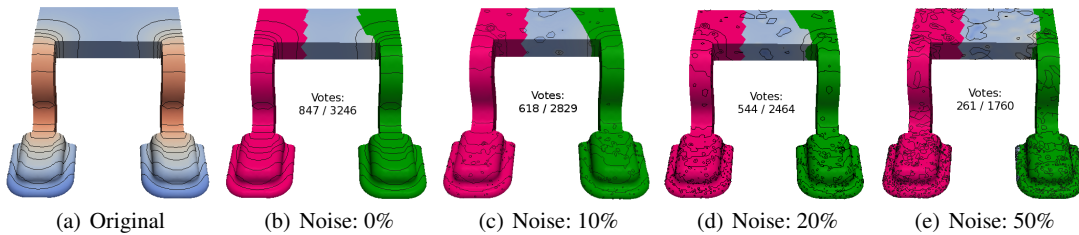


Fig. 13 Effect of noise on symmetry detection. (a) Original conductor dataset. (b) to (e) The results obtained for different levels of noise. A few level sets are shown to show local effect of noise. Number of votes for this symmetry out of total votes is also mentioned.

5 Performance and Robustness

Table 2 lists the runtimes of various stages of the pipeline for some of the datasets. All experiments were conducted on a workstation with a dual core Intel Xeon 2GHz processor, 4GB, and on Java Runtime Environment JRE-6. As evident from results shown, the time taken for the sampling, LFD computation, pairing, and voting stages is small. The overall

time is primarily determined by time taken in the clustering and region growing stages.

Noise in the data adversely affects the method as it based on comparing local properties, which are sensitive to noise. However, the clustering stage is able to detect significant symmetries even in presence of tolerable noise. We demonstrate the robustness of the symmetry detection pipeline with the following experiment. Given an input dataset, we introduce increasing levels of noise to generate synthetic

Table 2 Performance Results

Dataset		No. of Vertices	Time taken (seconds)		
			Early stages	Clustering	Region growing
Synthetic-1	Fig 5(a)	40000	0.37	0.28	0.57
Hydrogen	Fig 7(f)	16384	0.81	1.51	1.38
vonKarman	Fig 7(i)	80000	8.43	2.21	1.97
Synthetic-2	Fig 10(a)	40000	2.72	8.21	1.31
Conductor	Fig 7(b)	8598	0.56	0.59	6.11
2OAU	Fig 8(a)	13494	0.53	0.23	1.27
1GRM-1	Fig 8(c)	6210	0.57	0.68	2.66
1GRM-2	Fig 8(f)	25943	1.11	0.22	1.12
Heater	Fig 11(a)	28675	1.18	2.24	7.22
Water	Fig 9(a)	262144	12.59	99.91	11.91
CryoEM-1	Fig 9(i)	68921	5.97	2.98	2.87
CryoEM-2	Fig 9(k)	132651	6.22	6.82	7.75
Vortex	Fig 9(m)	262144	3.26	2.69	11.42

datasets. Figure 13 shows results of an experiment on the conductor dataset with noise level specified by a parameter k that ranges from $k = 0$ to $k = 50\%$. Scalar values at a random set of $k\%$ points from the mesh are perturbed by a random value drawn from Gaussian distribution with zero mean and 10% standard deviation. All parameters used in the symmetry detection pipeline remain unchanged. The symmetric legs are detected in all the cases. As expected, the number of votes for the symmetry decreases with increasing noise levels. However, the number of votes are sufficient for the detection of translational symmetry in all cases. Further, the noise does affect local properties like contour curvatures and gradients. However, the clustering stage tolerates this noise and detects the symmetries. The number of votes for a symmetry may go down below the detectable threshold. In such cases, the pairing tolerance parameter may be tuned to increase the votes. However, determining a good tolerance parameter is an interesting and challenging problem. We also suggest employing a preprocessing routine to remove outliers and smoothing the data for better results.

6 Conclusions and Future Work

In this paper we have defined the notion of symmetry in scalar fields which incorporates geometry and proposed a voting based approach for detecting partial and approximate symmetry. This approach overcomes the problems associated with detecting symmetry using similarity within topological structures like contour trees since they do not capture geometry. Our method works for scalar fields defined on general 2-manifolds including 2D Euclidean space and 3D Euclidean space. We show results for variety of datasets from different research domains including real, synthetic, and those obtained from simulations, thus showing the versatility of our method.

One major disadvantage of the method is efficiency. In future, parallelizing various stages of the symmetry identification pipeline can be done to increase efficiency. Another limitation is that currently only translation and rotation transformations are considered. Supporting all transformations is a much harder problem. The technique of voting in transformation space and clustering may not work in such a scenario, and a different approach may be required. We plan to address these challenges in future.

Acknowledgements This work was supported by a grant from Department of Science and Technology, India (SR/S3/EECE/0086/2012) and by the Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science.

References

- Alliez, P., Cohen-steiner, D., Devillers, O., Levy, B., Desbrun, M.: Anisotropic polygonal remeshing. *ACM Transactions on Graphics* **3**, 485–493 (2003)
- Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. *Comput. Geom.* **24**(2), 75–94 (2003)
- Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Analysis of scalar fields over point cloud data. In: *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '09*, pp. 1021–1030 (2009)
- Cohen-steiner, D., J.M.-Morvan: Restricted delaunay triangulations and normal cycle. In: *Proc. of Symposium on Computational Geometry*, pp. 312–321 (2002)
- Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619 (2002)
- Correa, C.D., Lindstrom, P., Bremer, P.T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Trans. Vis. Comput. Graph.* **17**(12), 1842–1851 (2011)
- Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining* (1996)
- Gyulassy, A., Natarajan, V.: Topology-based simplification for feature extraction from 3d scalar fields. In: *Visualization, 2005. VIS 05. IEEE*, pp. 535–542. IEEE (2005)
- Hong, Y., Shen, H.W.: Parallel reflective transformation for volume data. *Computer and Graphics* **32**(1), 41–54 (2008)
- Kazhdan, M.M., Chazelle, B., Dobkin, D.P., Funkhouser, T.A., Rusinkiewicz, S.: A reflective symmetry descriptor for 3d models. *Algorithmica* **38**(1), 201–225 (2003)
- Kerber, J., Bokeloh, M., M.Wand, Krüger, J., Seidel, H.P.: Feature preserving sketching of volume data. In: R. Koch, A. Kolb, C. Rezk-Salama (eds.) *Vision, Modeling, and Visualization* (2010)
- Kerber, J., Wand, M., Krüger, J., Seidel, H.P.: Partial symmetry detection in volume data. In: P. Eisert, K. Polthier, J. Hornegger (eds.) *Vision, Modeling, and Visualization* (2011)
- Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.* **21**(4), 163–169 (1987)
- Mitra, N., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics* **25**, 560–568 (2006)
- Mitra, N.J., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3D Geometry: Extraction and Applications. In: *EG 2012 - State of the Art Reports*, pp. 29–51 (2012)

-
16. Thomas, D.M., Natarajan, V.: Symmetry in scalar field topology. *IEEE Transactions on Visualization and Computer Graphics* **17**(12), 2035–2044 (2011)