Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

AMAL ABOULHASSAN, RONELL SICAT, DANIEL BAUM,
OLGA WODO AND MARKUS HADWIGER

Comparative Visual Analysis of Structure-Performance Relations in Complex Bulk-Heterojunction Morphologies

The manuscript will appear in a slightly revised version in a special issue of Computer Graphics Forum.

ZIB Report 17-16 (April 2017)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Comparative Visual Analysis of Structure-Performance Relations in Complex Bulk-Heterojunction Morphologies

Amal Aboulhassan, Ronell Sicat, Daniel Baum,
Olga Wodo and Markus Hadwiger

Abstract

The structure of Bulk-Heterojunction (BHJ) materials, the main component of organic photovoltaic solar cells, is very complex, and the relationship between structure and performance is still largely an open question. Overall, there is a wide spectrum of fabrication configurations resulting in different BHJ morphologies and correspondingly different performances. Current state-of-the-art methods for assessing the performance of BHJ morphologies are either based on global quantification of morphological features or simply on visual inspection of the morphology based on experimental imaging. This makes finding optimal BHJ structures very challenging. Moreover, finding the optimal fabrication parameters to get an optimal structure is still an open question. In this paper, we propose a visual analysis framework to help answer these questions through comparative visualization and parameter space exploration for local morphology features. With our approach, we enable scientists to explore multivariate correlations between local features and performance indicators of BHJ morphologies. Our framework is built on shape-based clustering of local cubical regions of the morphology that we call patches. This enables correlating the features of clusters with intuition-based performance indicators computed from geometrical and topological features of charge paths.

1 Introduction

Organic photovoltaic solar cells (OPVs) are a promising flexible low-cost alternative to traditional solar cells. An OPV is a device composed of three main parts (Fig. 1): the two electrodes (anode and cathode), and a layer sandwiched between the electrodes called the Bulk-Heterojunction (BHJ) [29]. The BHJ is a blend of two materials: the *donor* and the *acceptor*. Donor and acceptor are separated by a surface called the *interface*, whose morphology significantly influences the overall performance of BHJ materials.

Fig. 1 illustrates the photovoltaic process that occurs in a sequence of stages: photon absorption, exciton generation, exciton diffusion, charge separation, and charge transport. At each stage of the photovoltaic process, its performance is critically affected by the morphology of the BHJ. The objective of designing good BHJ materials is to maximize the generated photoelectric current.

However, the analysis workflow currently employed in practice for finding optimal BHJ structures still solely depends on the use of global statistics. First, scientists create simulators for the BHJ generation process. These simulators usually produce synthetic

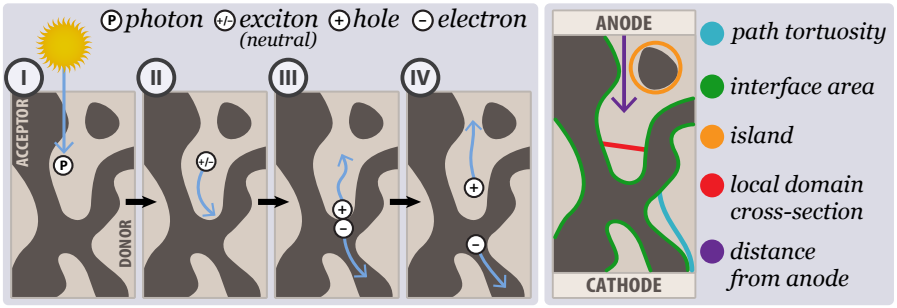

Figure 1: Illustration of the four stages of the photoelectric current generation process in a BHJ morphology: I. photon absorption, II. exciton generation, III. exciton diffusion, IV. charge separation and transport. Right: Structural features that influence the quality of these stages, ultimately affecting the efficiency of OPV solar cells.

BHJ volumes in the form of regular scalar fields. From these volumes, structure metrics such as domain size and acceptor-donor interface area are computed. Finally, global statistics over the entire morphology are computed to characterize it with respect to performance. However, performing comparisons via global statistics only has limitations:

- Global statistics ignore the dependency among the different stages of the photoelectric current generation. For example, a local region might have features with excellent exciton diffusion probability. However, at the same time it might also have features that result in low photon absorption probability. In this case, it is very likely that no excitons will be generated at all. In global statistics for exciton diffusion, however, these "dead zones" will have the same contribution as active regions with many excitons.

- Designing efficient BHJ materials requires simulating and comparing hundreds of different possible morphologies with dozens of simulated fabrication parameters each, while understanding their local performance characteristics and their correlations with local morphology and simulation parameters. Global computations alone cannot uncover these correlations. A detailed comparison requires additional local visual analysis.

We address these limitations with the following contributions:

- We facilitate local analysis based on the concept of *patches*. Patches are small, local, cubical regions in a BHJ morphology. For each BHJ volume, a set of patches is computed whose locations and sizes are adapted to the local morphology (Fig. 3).

- We cluster patches according to *patch descriptors*. We define a local curvature distribution descriptor, the *patch interface shape distribution (PISD)* of the acceptor-donor interface contained in a patch. For each patch, we also compute *performance indicators*, for example from local charge path computations.

- We provide practitioners in BHJ design with a tool for interactive visual exploration, analysis, and comparison of performance-structure correlations and the

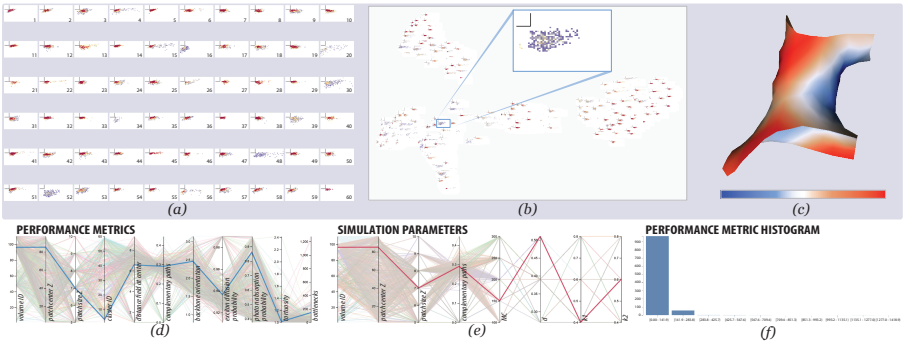

Figure 2: Our visual analysis workflow comprises several linked views: The set of patch descriptors (curvature distributions) is visualized after (a) k-medoids clustering, and (b) t-SNE dimensionality reduction and visualization. (c) Interface surface of a selected patch color-coded by Gaussian curvature. (d,e) Parallel coordinates views for (d) performance metrics, and (e) simulation parameters. The highlighted lines represent the currently selected patch. (f) Histogram of a selected performance indicator after filtering in the parallel coordinates views.

impact of BHJ simulation parameters, using brushing and linking of different local and global parameters. This enables scientists to relate local parameters to global performance. We provide a set of 2D and 3D visualization modules designed for local analysis of BHJ morphologies.

Using our visualization tool (see Fig. 2), scientists can interactively explore local regions (*patches*) and their charge properties. Instead of comparing global statistics only, they can compare *sets of patches* across morphologies, based on both visual exploration and interactive queries. The queries allow scientists to visualize ranges of parameters as well as the dependencies between them.

Our results demonstrate how this has enabled scientists to analyze and understand local BHJ properties and their relation to photoelectric current generation with higher confidence. Moreover, they can interactively perform a top-down analysis of a set of hundreds of global morphologies to the 3D information of a specific region in a specific morphology. This helps them choose better fabrication parameters and design strategies for the BHJ fabrication.

2 Related Work

2.1 Analysis of Bulk-Heterojunction Materials

Previous approaches for the analysis of BHJ materials still largely depend on trial and error, and on comparing global statistics only. To make this process more efficient, Wodo et al. [30] developed an approximation model that extracts a representative set of charge paths that are physically intuitive. This approach has proved successful in simplifying complex simulations. As a result, it has started to be employed in the analysis of simulations [11].

Another example of computational methods is discussed by Ray and Alam [22]. In this research, they generated tens of morphologies and explored if randomness influ-

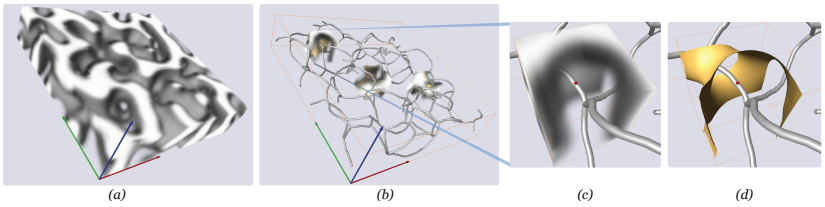

Figure 3: **Patches** are small, local, cubical regions that are extracted from the whole BHJ morphology (a) for local analysis. Patches are sampled along the backbone (b) of the BHJ morphology. This allows capturing more informative samples compared to sampling patches on a regular grid as well as adapting patch size to the local size of the morphology. In (b) three example patches are shown via volume renderings embedded in the backbone of the morphology. Patches can be inspected via (c) a close-up view of a selected patch, showing the center of the patch (red dot) on the backbone, and (d) a visualization of the interface between acceptor and donor contained in the patch.

ences the performance. They also compared the performance to regular shape structures.

These techniques have obvious limitations. Overall BHJ performance results from the complex interplay between many features related to structure and performance. However, current techniques allow for comparing only a few at a time. Therefore, finding the best-performing structures is an open question [22]. All these techniques also depend on global statistics, which hampers exploring important geometric features [1]. Global techniques also do not allow relating local dependencies to global results.

In this paper, we tackle these limitations by enabling the comparison of local features for hundreds of morphologies at the same time. We use the concept of visual exploration to make this comparison effective and give fast feedback about the simulation parameters that lead to certain performance characteristics.

Employing skeletonization for generating the *backbone* of a BHJ morphology has proved to be a useful tool that facilitates local characterization of BHJs [1]. We also compute backbones, using a conventional thinning algorithm [21], based on distance maps [15] computed for each morphology. Other simplified representations include Reeb graphs [19], distance field-based methods [9], extremum graphs [17], and other topological methods [13]. We chose the backbone model because it facilitates visual connectivity exploration in 3D.

2.2 Shape and Curvature Analysis

Several methods have been proposed to represent shapes by geometric invariants. For example, Reuter et al. [23] proposed Shape-DNA, which is based on the computation of the eigenvalues of the Laplace-Beltrami operator. They showed that this signature is very well suited to identify equal shapes, but also to group similar shapes. Sun et al. [25] proposed a point signature based on the properties of heat diffusion, which they call Heat Kernel Signature (HKS). Bronstein and Kokkinos [4] extended this signature to a scale-invariant version (SI-HKS). They showed that their method is superior to both Shape-DNA and HKS in comparing shapes from the Shape Google database [3].

Curvature as a major shape cue has proven useful in physical applications that need to characterize microstructure. Our *patch interface shape distribution* (PISD) as

a descriptor for local patch shape is inspired by the work of Chen et al. [6], where they introduced an interface shape distribution (ISD) to globally characterize and compare different morphologies of nanoporous gold. We extend the ISD concept from a global to a local characterization of shape by computing one ISD for each patch. Curvatures are computed from the local morphology's acceptor-donor interface. We favored this curvature-based approach since the interface curvature can be directly influenced by morphology simulation parameters. Furthermore, Shape-DNA as well as HKS-based approaches were primarily developed to compare the global structure of shapes. In contrast, curvature is a very local feature. It is also unclear how well the other methods can deal with the patch boundaries we introduce.

To decrease the resulting complexity of thousands of PISD descriptors to a manageable amount, such that they can be explored interactively, we perform dimensionality reduction and k-medoids clustering (using the medoid instead of the mean as cluster center). t-SNE [27], which is an improved version of stochastic neighbor embedding [12], is used to map many PISDs into two-dimensional space such that local distances are preserved. A hierarchical variant of t-SNE was recently also developed [20].

Kindlmann et al. [16] described curvature-based transfer functions. We also visualize curvature of the acceptor-donor interface. However, our PISD descriptor characterizes an entire local neighborhood (a patch) instead of only a point on the surface. Xia et al. [32] proposed curvature-based analysis of protein electrostatic surfaces to explore correlations between metrics derived from the principal curvature and protein functions. Soldea et al. [24] used a curvature-based segmentation approach to identify planar or cylindrical iso-surfaces, and volumetric regions with saddle-like iso-surfaces. Explicit shape identification of local regions has also been applied, for example ellipsoidal Gaussian models for identifying fibre shapes in laser scanning confocal microscopy [14].

2.3 Ensemble Visualization

Ensemble visualization techniques are important when multiple runs of a simulation are computed. Similarly, we cluster and compare patches from morphologies that result from simulation runs with different parameters. An important issue in ensemble visualization is characterizing the statistical distribution of the ensemble members around the most central, or "deepest," member, for which techniques such as contour boxplots [28] and curve boxplots [18] have been developed. Demir et al. [7] proposed to use a locally most-representative shape together with a region-wise centrality quantification, inspired by previous work on contour variability plots [8]. Computation of this most-representative shape, however, requires shapes to be superimposed as well as having similar geometries. It is not clear how this can be achieved for our patches. Instead, in our framework each cluster is represented by the medoid of the PISD descriptors in the cluster.

2.4 High-Dimensional Parameter Space Exploration

Finding relationships between simulation parameters and the corresponding simulation results is a very challenging problem, especially due to the high dimensionality of the involved parameter spaces. Visualization can help reduce the parameter space into smaller selections, and hence reduce extensive trial and error.

Bruckner and Möller [5] proposed a framework for physically-based simulation that enables graphics artists to decide on a set of simulation parameters that give de-

sirable effects for phenomena such as fire, smoke, etc. They proposed density-based clustering of animation sequences into fewer segments for a more concise view. They represent the simulation parameters using star plots that show representative frames for linked clusters.

Beham et al. [2] proposed the Cupid system for geometry generators used in video games and computer vision applications. Cupid helps select geometric generator parameters that avoid generating redundant or invalid geometric shapes. The system starts with a hierarchical DBSCAN clustering to group the 3D shapes based on mesh similarity. The cluster hierarchies are visualized using radial trees, circular dendrograms, and treemaps. Parameter selection is enabled through composite parallel coordinates.

Torsney-Weir et al. [26] proposed the Tuner system for parameter finding for image segmentation. Tuner proposes a Gaussian model for generating sample points. The samples and parameters are visualized using a combination of visual views: Pareto panel, response view, controls, and histograms.

3 Overview

Our visual analysis workflow is based on extracting a large number of small local regions from each BHJ volume. We call those regions *patches*. Two main factors influence the local contribution of a patch to the overall performance of the morphology: acceptor-donor interface *shape* and *topology*. Many of these properties intuitively map to the efficiency of photoelectric current generation. Based on these observations, we characterize the *interface shape* using each patch's curvature distribution (PISD), leverage topological information from the *backbone* of the morphology, and compute a variety of *performance indicators* estimating BHJ efficiency.

Patch sampling. For simplicity, we use cubical patches (e.g., 9^3 voxels), although other patch shapes could be considered as well. Each patch size is adapted to the local morphology. Instead of extracting patches from the volume by sampling a regular grid, we sample along the backbone of the BHJ morphology. From each patch we then compute representative local properties (see Fig. 3).

Patch descriptors. For each patch, we compute a patch descriptor that is used for local characterization. In collaboration with BHJ scientists, we have defined the patch interface shape distribution (PISD), which is based on acceptor-donor interface curvature computations, as an efficient tool for characterizing local shape.

Performance indicators. Table 1 summarizes the performance indicators that we are using. These indicators have complex interdependencies, and in many cases they correspond to conflicting design properties. Therefore, in order to design a BHJ morphology, scientists need to explore combinations of these indicators based on the optimal choice of interdependent parameters. The interdependency of all performance indicators, as well as the influence of local shape, represents the biggest challenge for OPV designers. We tackle this challenge by providing concise visualizations of thousands of patches and allowing scientists to specify visual queries about structure-performance relationships.

Dimensionality reduction and clustering. In order to be able to handle a large number of patches from many different morphologies, we first perform dimensionality

reduction of the PISD descriptors using principal component analysis (PCA), and then perform k-medoids clustering to group similar patches together.

t-SNE visualization. In order to be able to visualize the high-dimensional space corresponding to the PISD descriptors, we employ t-SNE [27] to visualize all PISD descriptors as "icons" in a two-dimensional space that scientists can easily interact with. This is illustrated in Fig. 2 (b) and Fig. 4.

Visual analysis framework. Our framework allows scientists to interact with patches by clicking on PISD icons in either the cluster view (Fig. 2 (a)) or the t-SNE view (Fig. 2 (b)). The corresponding 3D morphology is then visualized on demand, and individual patches can be inspected. Performance indicators and simulation parameters are visualized in parallel coordinates views (Figs. 2 (d) and (e)). More details can be inspected in histograms (Fig. 2 (f)). All of these views allow linking and brushing.

descriptor / indicator	description	influence on physical properties
PISD	patch interface shape distribution (curvature distribution) of the interface	bottlenecks and amount of excitons harvested by the interface
Abs	distance from a point in the donor to the anode	photon absorption probability
Diff	distance from a point in the donor to the nearest point on the interface	life time of the exciton, and exciton diffusion probability
Orient	angle between the backbone and the z-axis (anode to cathode)	bottlenecks and tortuosity
Tort	(hole) path tortuosity	life time of (hole) charges
BN	(hole) path bottleneck	(hole) charge transportation speed
Comp	regions that are connected to both electrodes at the same time (complementary paths)	charge accumulation

Table 1: **Patch descriptors/performance indicators** help map structure and topological features to photoelectric current quality.

4 Patch Descriptors and Performance Indicators

The patch descriptors (PISD), the performance indicators, and k-medoid clustering are computed in a pre-processing stage in order to prepare for the subsequent interactive visual analysis.

4.1 Morphology Backbone (Topology)

We start by generating the *backbone*, i.e., the skeleton of the BHJ morphology, using the algorithm proposed by Aboulhassan et al. [1]. We chose this representation because it provides an abstraction of the connectivity between local regions of the morphology as well as of the distances to the acceptor-donor interface.

In order to estimate properties of charge paths, we compute shortest paths in the morphology, following a previously proposed model [29]. Two types of paths are computed: paths from the interface to the anode via the donor: *hole paths*; and paths from the interface to the cathode via the acceptor: *electron paths*.

4.2 Patch Sampling

In order to successfully capture local features, the selection of the patches needs to satisfy the following conditions: (1) The size of each patch should not be too big, to be local, and not be too small, to capture enough interface fragments (see the discussion below). (2) Patches should be sampled at locations that are neither too close to each other, to avoid redundancy, nor too far from each other, to avoid missing important features.

Patch centers. Since the backbone lies, by definition, in the middle of neighboring interface fragments, we can use it to guide the determination of patch centers. We select all patch centers to lie on the backbone. Currently, we consider only the donor because it includes all the charge transport stages needed in the current analysis. The patch centers are distributed iteratively along the backbone such that the patches cover the whole backbone without creating too much overlap. That is, we use the minimal number of patches that guarantees coverage of the backbone.

Patch sizes. To determine the size of each patch, we compute a distance field from the backbone to the interface. The value of the distance field at each patch center on the backbone then determines the patch size via the following equation:

$$D = s \cdot d(\mathbf{x}) + \Delta D. \tag{1}$$

Here, $d(\mathbf{x})$ is the value of the distance field at the patch center, s is a constant scaling factor, and ΔD is a fixed padding of extra voxels that ensures that enough interface fraction is included. The local patch size is then set to D^3 voxels. The parameters s and ΔD are chosen by the user. In the current study, $s = 1$ and $\Delta D = 4$ were adequate for our exploration and our collaborators.

4.3 Patch Interface Shape Distribution (PISD)

Inspired by previous work in physics/materials science on the characterization of nanoporous gold [6], we define a patch descriptor that characterizes local shape via its curvature distribution.

Our *patch interface shape distribution (PISD)* descriptor is computed as the probability distribution of principal curvature values κ_1 and κ_2 (maximum and minimum

principal curvature, respectively) of the acceptor-donor interface surface. More precisely, the PISD gives the probability of finding a small patch of interface surface with a given pair of principal curvatures (κ_1, κ_2).

Curvature normalization. In order to be able to meaningfully compare different patches and different morphologies, it is crucial that the curvature values are normalized in a suitable way. We achieve this by normalizing with the physically meaningful measure of *surface area per volume* [6], denoted by S_v below.

In order to obtain a probability density, the surface area for each combination of (κ_1, κ_2) must also be normalized with respect to the total interface area in the given patch, denoted by A_{patch} below.

PISD definition. We thus define the PISD descriptor as a probability density over the two-dimensional domain $\left(\frac{\kappa_1}{S_v}, \frac{\kappa_2}{S_v}\right)$, given by

$$PISD\left(\kappa_1/S_v, \kappa_2/S_v\right) := \frac{\mathrm{d}^2 A\left(\kappa_1/S_v, \kappa_2/S_v\right)}{\mathrm{d}\left(\kappa_1/S_v\right)\ \mathrm{d}\left(\kappa_2/S_v\right)\ A_{patch}}, \tag{2}$$

where A_{patch} is the total surface area of the interface contained inside the patch. See Fig. 4 for a visualization of an example PISD.

We implement this function in discretized form as a simple 2D array, where each bin contains the probability that a randomly chosen point on the interface maps to the range $\left(\frac{\kappa_1}{S_v} + \Delta\frac{\kappa_1}{S_v}, \frac{\kappa_2}{S_v} + \Delta\frac{\kappa_2}{S_v}\right)$, where $\left(\Delta\frac{\kappa_1}{S_v}, \Delta\frac{\kappa_2}{S_v}\right)$ is the size (κ-range) of the bin.

The *surface area per volume* measure S_v is computed as

$$S_v := \frac{A_{patch}}{V_{donor}}, \tag{3}$$

where V_{donor} is the total volume of the donor part inside the patch. We refer to previous work for details on S_v [6].

Surface and curvature computation. In order to perform the computations required above, we obtain the interface surface inside a patch via *Marching Cubes* followed by mesh refinement. We estimate the curvatures on the obtained high-resolution triangle mesh.

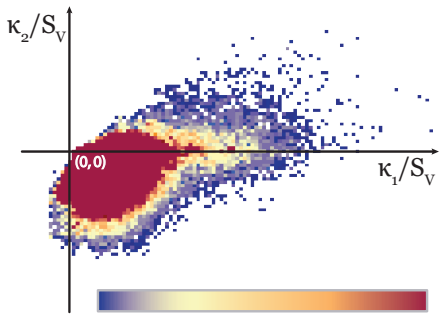

Figure 4: **PISD descriptor.** Each patch, set of patches, or whole morphology, can be visualized via the corresponding patch interface shape distribution (PISD) in the domain of normalized principal curvatures. This helps with understanding shape characteristics and similarities/differences between patches/morphologies.

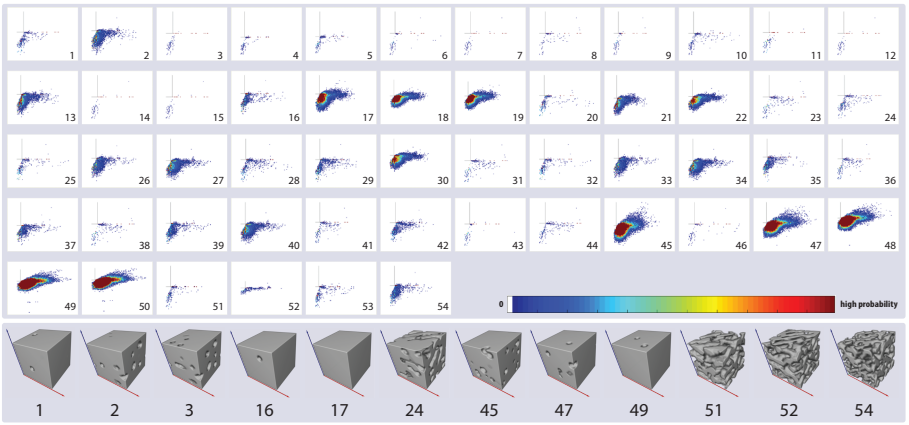

Figure 5: **Comparative visualization of patch interface shape distributions.** We show the overall PISD of each of the 54 morphologies used in this study, and selected examples of the corresponding volumes (bottom row). By studying the PISDs, scientists can decide to quickly filter some volumes—for example, less dense volumes. The PISD icons here show the total distribution of all patches in each morphology.

Dimensionality reduction and clustering. In order to handle large numbers of patches, each PISD is treated as a point in a high-dimensional space by simply mapping the discretized 2D domain to a 1D vector whose dimensionality is the number of PISD bins. In this space, we then perform dimensionality reduction using principal component analysis (PCA). Afterward, similar patches are identified via k-medoids clustering, which identifies meaningful groups of patches that have similar curvature distribution, i.e., similar PISD descriptors, and hence have similar morphology with respect to shape.

4.4 Performance Indicators

In this section, we explain the functions used to compute the performance indicators discussed in Table 1. We characterize the performance of a patch using the indicators proposed by Wodo et al. [31] restricted to each patch. We also include two indicators of high interest to the scientists: (1) the *bottleneck* indicator, which estimates the charge transport bottlenecks and their correlation to local cross-sectional areas; (2) the *exciton diffusion* indicator, which estimates the correlation between the distance at the backbone to the nearest interface fragment and the corresponding probability of exciton diffusion to the interface. Both indicators are discussed in detail in previous work [1].

In addition to these performance indicators, we propose the following additional indicators to cover more stages of the photoelectric current generation process.

Photon absorption. The first step of the photovoltaic process is photon absorption, which usually happens in the donor:

$$Abs = e^{-h(\mathbf{x})/H_d}, \tag{4}$$

where $\mathbf{x}$ is the current voxel, $h(\mathbf{x})$ is the physical distance from $\mathbf{x}$ to the anode, and H_d is the absorption coefficient, which is a property of the material. We use $H_d = 100$ in the current paper.

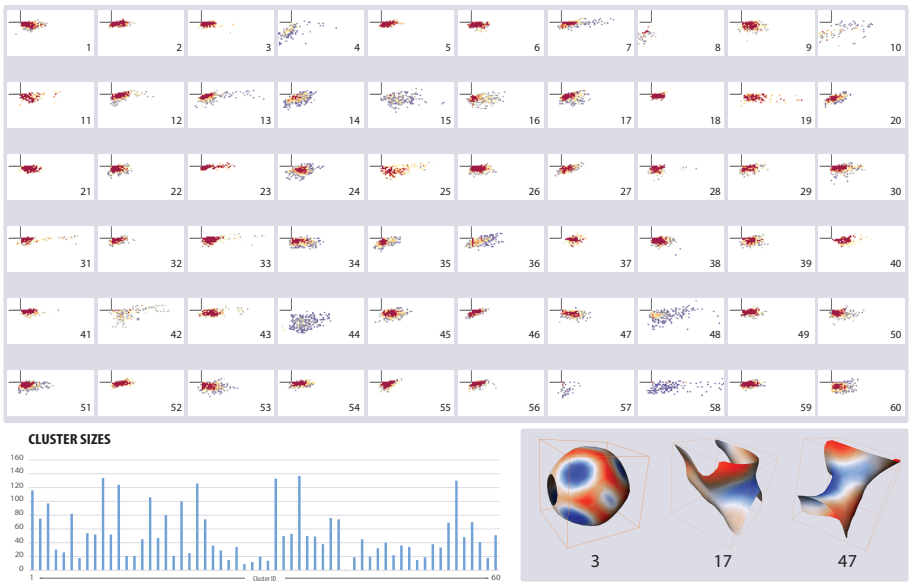

Figure 6: **Patch cluster representatives.** The cluster centers (medoids) of the 60 patch clusters (computed from 3,135 patches) used in our case studies. 3D views of example patches are shown in the bottom right. From the cluster centers, scientists can obtain a quick characterization of the major shapes generated by the corresponding simulation parameters with respect to curvature (PISD) characteristics.

In the current study, the scientists chose to use the $h(\mathbf{x})$ value at the patch center to compute the corresponding *Abs* value as given above as the representative value for a given patch to be used in the correlation analysis.

Tortuosity. We use the indicator proposed by Wodo et al. [31]. To compute the tortuosity of a path, we first determine the length L of the shortest path from a given point in the donor to the corresponding electrode, i.e., the anode. In this way, we estimate the deviation of the shortest path from the ideal shortest path (a straight line of length Z)

$$Tort = \frac{L}{Z}. \tag{5}$$

The *Tort* indicator for each patch is computed in the same way, but only in a segment of the whole path clipped to the boundaries (the bounding box) of the patch.

In the current study, the scientists chose to use the maximum tortuosity value of all path segments passing through each patch as the representative *Tort* value to use in the correlation analysis.

Complementary paths. We propose the following indicator for estimating hole/electron balance at each patch. This indicator is computed by first counting the number of voxels that belong to the patch interface that are the origin of paths to both the anode and the cathode, respectively. We then normalize this value by dividing it by the total number of voxels that belong to the interface inside the patch.

5 Visual Analysis

After the patch descriptors and performance indicators have been computed, visual analysis can be performed interactively. We describe the main views that comprise our visualization system.

5.1 Patch Cluster View

The aim of this view is to provide a concise visualization of all patch clusters. As described above, all patches are clustered based on the corresponding PISD descriptors, which is an important property of interest for visualization and analysis. In this view, we represent each patch cluster by the PISD icon of the cluster center.

Due to the use of k-medoids clustering, the cluster center is a patch that actually occurs in the cluster, and we can therefore simply use the descriptor of this patch. An example PISD icon is illustrated in Fig. 4. All path cluster PISD icons are then visualized in a compact 2D layout. Each icon can have a label of interest. In this paper, we use the cluster ID by default, and allow the user to add arbitrary text labels. This view is illustrated in Fig. 2 (a).

5.2 t-SNE View

In addition to the patch cluster view described above, where the layout of cluster centers is essentially arbitrary, we provide a second view for the visualization of clusters and patches, using t-SNE [27] to obtain a meaningful 2D layout that groups similar patches/clusters together in the 2D visualization domain (see Fig. 2 (b)).

The t-SNE view can be used to very effectively inspect similarities between clusters, similarities between individual patches, as well as for visualizing the contents of a selected cluster. We provide zooming functionality to allow one to view all clusters but also to investigate specific regions in more detail. When all patches/clusters are being viewed, the plots might not be readible anymore. For this, we implemented a lense functionality when hovering with the mouse over a specific patch plot.

5.3 Performance Metrics View

Performance metrics are displayed as parallel coordinates as illustrated in Fig. 2 (c). We chose parallel coordinates because they are efficient at visualizing dependencies between parameters, which is one of the main goals of this paper.

This view is not only important for visualization but also for queries: Users can choose a range of performance metrics of interest and instantly see correlated values. Similarly, multiple queries among dependent variables can be performed.

This view is linked to the patch cluster view. Users can brush (select) in the performance metrics view and apply the corresponding query in the patch cluster view, or they can click on one PISD icon and see the corresponding line highlighted in the performance metrics view, as shown in Fig. 2 (d).

5.4 Simulation Parameters View

Similar to the performance metrics view, the simulation parameters view uses parallel coordinates for the exploration of simulation parameters. This view is linked to both the performance metrics view and the patch cluster view. It is illustrated in Fig. 2 (e).

5.5 Performance Metric Histogram View

After performing queries via brushing, sometimes the corresponding relationships are not obvious since there are lines in the parallel coordinates views that are passing through all values. This may give an indication that there are no correlations after applying this query. By exploring the histogram of the filtered data, this conclusion can be corrected if there it is found that some bins have higher values than other bins, as shown in Fig. 2 (f).

5.6 3D Exploration

Acceptor-donor interface surface view. Our collaborating scientists find it helpful to be able to look at what interface surfaces look like in 3D, to be able to think about the corresponding curvature distributions and how they are influencing the charge paths. To visualize this data, we allow users to select any patch in the patch cluster view, and then display the triangle mesh of the corresponding interface surface, as shown in Fig. 6 (bottom right).

Pathlines view. The scientists want to be able to visualize all charge paths that are passing through any given patch, in order to get more contextual information on why there are high or low bottleneck values, and in order to understand specific tortuosity values. Our pathlines view can therefore both visualize all pathlines selected by a query as well as all pathlines that are passing through a given patch of interest. In addition, we offer the possibility to display only the segments of paths inside a given patch, in order to explore information such as local tortuosity. All paths are visualized using ray-casting of surface-shaded tubes. We also allow color-coding of each path by its tortuosity value, as shown in Fig. 7.

Backbone view. The scientists consider the backbone view to be one of the most important basic views of our framework, since it gives a good uncluttered summary of the topology of the morphology (its connectivity), and its relationship to individual patches, as shown in Fig. 3.

6 Evaluation

We have evaluated our system together with our collaborators who are experts in BHJ design, analysis, and fabrication.

System setup. We have used the Ising model-based simulator proposed by Heiber and Dhinojwala [11] to generate a total number of 500 different morphologies. All simulations were run on a compute cluster with 342 nodes of 8 core Intel Xeon X5570 (2.93GHz) and 24 GB memory, running Red Hat Linux. The simulator we used [10] is implemented in C/C++.

As our basic visualization system we use Amira, for which specialized plugins were implemented. The cluster view and parallel coordinates were implemented using D3.js. K-medoids clustering and PCA computations were implemented using Matlab. The shortest paths computations were implemented using the boost C/C++ graph library. Shortest path computations, preprocessing and visualization are all run on a machine with Intel Xeon CPU E5-2698 v3 @ 2.30GHz (2 processors) with 128 GB RAM.

The simulation to compute all morphologies was run overnight. The sizes of each of the 500 final morphologies range between 1M to 6M voxels per morphology.

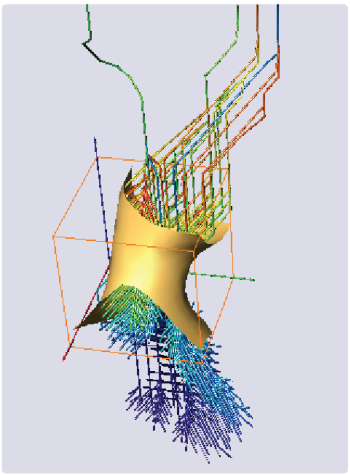

Figure 7: **Charge paths traversing a selected patch.** The charge paths passing through any given patch can be inspected individually. Important path properties, such as their (global or local) tortuosity, can be color-coded on the visualization of the path line.

6.1 Case Studies

In order to evaluate the usefulness of our system, we asked scientists with expertise in computational physics, microstructure analysis, and OPV solar cell design to perform visual exploration and analysis using our system. In this paper, we will discuss only two of the main findings due to the limited size of the paper.

Morphology generation. We started by sampling meaningful ranges for all simulation parameters for morphology generation, in accordance with the experience of our collaborators. Using these parameter settings, we generated 500 different morphologies.

Using our system, we then first performed a selection of a target subset for more detailed analysis together with the scientists, using the PISD descriptors of entire morphologies, until we reached a set of 54 target morphologies that they thought cover different shape- and topological features of interest sufficiently well. The PISD histograms of these 54 morphologies are depicted in Fig. 5.

Patch computation. From the 54 target morphologies, we computed a total of 3,135 patches sampled along the backbones of the morphologies. For each patch, the corresponding PISD descriptor was computed. Using a resolution of 10×10 bins per PISD, we obtained 100-D vectors. On these vectors we computed PCA for dimensionality reduction and retained the 6 largest PCA vectors after inspecting the corresponding eigenvalues, i.e., the variances in decreasing order. We looked at the decrease of the eigenvalues, as well as at the *explained variance* measure, which measures how much of the total variance is explained by the n largest eigenvalues.

Patch clustering. We then computed the patch clustering, using k-medoids clustering with $k = 60$ clusters, where k was determined empirically. The corresponding centers of these 60 clusters are shown in Fig. 6. These centers represent the 60 groups of patches used to classify the whole set of 3,135 patches with respect to their local shape characteristics (their PISD descriptors).

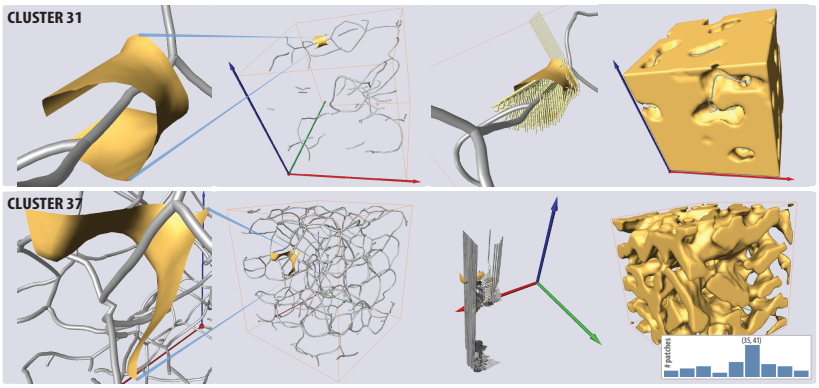

Figure 8: **Case study 1.** 3D views of patches that belong to clusters 31 and 37 as examples for the output generated during the analysis session for Case Study 1. This illustrates why these cases had low bottleneck values, although they are close to the anode. This is a new counterexample for a common domain expert belief. Moreover, they have some distribution of an "hour glass shape." For cluster 31, the figure shows that this is caused by the local topology, since few paths are allowed to pass through the patch, especially after branching.

6.1.1 Case Study 1: Finding low bottlenecks close to anode

In this case study, we start with the physical intuition of domain experts and progressively move toward a more detailed local analysis.

Basic intuition. Local shapes in the form of an "hourglass" increase the probability of bottlenecks compared to flatter shapes, which is a disadvantage. However, they have the advantage of having larger surface area than flat shapes, which is favorable for exciton disassociation (see Table 1). As a second consideration, the scientists had the intuitive belief that for patches that are closer to the anode, the probability of forming a bottleneck is higher than for patches farther away from the anode. At the same time, however, locations closer to the anode are favored by photon absorption.

Domain questions. (1) In order to analyze the impact of these conflicting considerations, the scientists posed the question whether there are patches with hourglass-like shapes that are close to the anode, but which at the same time have low (good) bottleneck values. (2) The second question was what the corresponding simulation parameters are that result in the generation of these properties. These questions cannot be answered using global statistics.

Visual analysis. Using our system, the scientists could answer their questions by first performing filtering in the parallel coordinates view to isolate patches with low bottleneck values and high photon absorption. Also, they restricted their selection to high complementary paths, to make sure that low bottlenecks are not due to "islands" (isolated parts of the morphology, see Fig. 1). Using this query, they filtered the selected PISD descriptors.

To determine correlations with clusters, they explored the performance metric histogram shown in Fig. 8 (bottom right). They noticed that a high number of patches comes from clusters 35 to 41. From the PISD descriptors of these patches (Fig. 6), they determined that these patches have high curvatures. Therefore, the scientists applied an additional query to select PISD descriptors with curvature probabilities corresponding

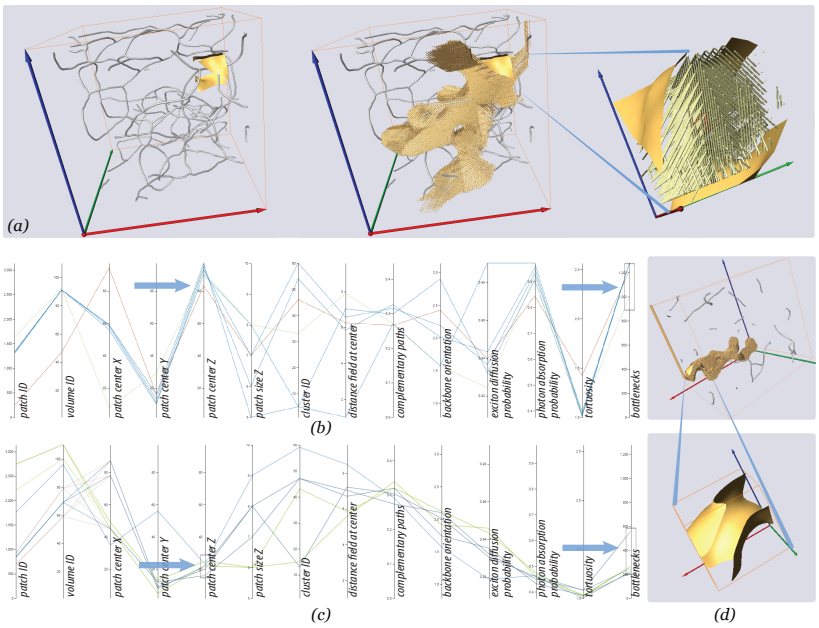

Figure 9: **Case study 2.** (a): Before analysis using our system, the domain experts considered a tortuosity value of less than 1.1 as good. Using the Case Study 2 analysis, however, they could strengthen this assumption by jointly exploring good tortuosity values and bad bottleneck values for each patch, rather than for the whole volume; then generate histograms corresponding to this combination of properties instead. (b,c,d): A second question concerned bottlenecks close to the cathode (low z values). (a) query on very high bottlenecks after filtering the 3,135 patches. (b) query on intermediate–high bottleneck values; (c) one patch that resulted from the intermediate bottlenecks at low z values.

to ridge shapes with strong curvature. Two example patches are shown in Fig. 8.

For the patch in cluster 31, the reason is the local topology: the backbone passing through the patch has some connectivity, but it is not connected to the deeper-lying parts of the morphology. This means fewer paths are passing through, which results in a lower path density. At the same time, other branches send charges into different directions, which again reduces the path density.

For the patch in cluster 37, the reason is that it is located close to domain boundary, where charge paths are coming in from fewer directions. This attracted the attention of the scientists to the fact that the direction orthogonal to the electrodes (z axis) is not the only major factor in morphology design, but that the horizontal directions (x and y) are more important than they had previously believed.

The scientists further explored the simulation parameters to understand how to re-generate these cases. One observation is that the patch from cluster 31 belongs to the morphology generated by simulation parameter *donor percentage* d = 0.39. The scientists had the common belief that the closer the percentage of donor to acceptor is to 0.5, the better the BHJ performance should be. This finding drew their attention to the fact that sometimes lower donor percentage values can still be good or even better than higher values.

6.1.2 Case Study 2: Finding low (good) tortuosity combined with high (bad) bottlenecks

Basic intuition. (1) Before our framework, our domain experts always considered a morphology with global average tortuosity less than or equal to a value of 1.1 as good. However, using our system they realized that they can create stronger characterizations by selecting those parts that are close to ideal local tortuosity and high bottleneck values at the same time, and create histograms of bottlenecks for closer inspection after filtering patches accordingly. (2) Another common belief among domain scientists was that bottleneck values are low close to the cathode.

Domain questions. (1) Therefore, the first target scenario was characterizing morphologies with respect to where good tortuosity and bad bottleneck values occur in the same location. (2) The second question was whether it is true that bottleneck values are indeed low close to the cathode (at low values of z). Answering these questions was enabled by our framework for the first time.

Visual analysis. Fig. 9 illustrates the output of this case study. Regarding domain question (1), the scientists are interested in understanding why there are bad bottlenecks combined with good tortuosity. From the 3D view in Fig. 9 (a) we can see the influence of the curvature distribution, as well as of the topology, by exploring all (global) paths that are passing through the corresponding patch, and only the (local) path segments that are inside the patch.

For domain question (2), Fig. 9 (b) shows a query on *very high* bottlenecks. None of the selected patches are close to the cathode (low z values), which agrees with the scientists' assumptions. However, by exploring *intermediate to high* bottleneck values in Fig. 9 (c), they found that still relatively high bottlenecks *do* occur close to the cathode (low z). Further exploration in 3D as in Fig. 9 (d) showed—considering the backbone—that the topology is the reason for this counterexample to their previous assumptions.

6.2 Design Lessons Learned

The design choices in the current application have gone through several iterations. In the beginning, we proposed visualizing patches by a representative 2D slice through the 3D morphology. The scientists did not find this helpful for getting a quick overview of features corresponding to the 3D structure. They then proposed that focusing on curvature information is more helpful, since it has an intuitive meaning for engineers in general and characterizes shape very well for their purposes. This led to our definition of the PISD descriptor.

For exploration and filtering, we were mainly relying on parallel coordinates. However, the scientists in some cases found that this alone does not reveal enough correlations. Introducing additional histogram views helped them to enhance their analysis.

As a general comment, the scientists found the backbone to be very helpful. They also found that our entire system design has a large potential to be used for the analysis of other microstructures in different applications, such as porous materials.

7 Conclusions and Future Work

In this paper, we presented the first visual analysis framework for identifying structure-performance correlations in large sets of simulated Bulk-Heterojunction morphologies

based on local analysis.

Based on clustering patches, we proposed a cluster view and a t-SNE view to help with exploring groups of similar patches. Linked to these views, we proposed a set of parallel coordinates to visualize performance and simulation parameters, and dependencies between them. Before the development of our framework, domain scientists depended on global numerical statistics to compare morphologies. Using our framework, domain scientists can now explore hundreds of morphologies with their multivariate features in the same view using a top-down exploration paradigm. They can also explore the performance at the local level which helps them determine the best-performing morphologies with more accuracy and more insight on why they are performing better, or worse, than other structures.

The next steps in the development of our framework will include the extension to the acceptor part of the morphologies, as suggested by the domain scientists. Furthermore, we plan to extend the system to include editing features to explore the correlation between certain modifications and the overall resulting performance.

Acknowledgments

This work was supported in part by King Abdullah University of Science and Technology (KAUST). The third and fourth author were supported in part by Global Collaborative Research, KAUST: CRG-1-2012-THO-015-ISU.

References

[1] Amal Aboulhassan, Daniel Baum, Olga Wodo, Baskar Ganapathysubramanian, Aram Amassian, and Markus Hadwiger. A Novel Framework for Visual Detection and Exploration of Performance Bottlenecks in Organic Photovoltaic Solar Cell Materials. *Computer Graphics Forum*, 34(3):401 – 410, 2015.

[2] Michael Beham, Wolfgang Herzner, M Eduard Gröller, and Johannes Kehrer. Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE transactions on visualization and computer graphics*, 20(12):1693–1702, 2014.

[3] A M Bronstein, M M Bronstein, L J Guibas, and M Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics*, pages 1–20, 2011.

[4] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1704–1711, 2010.

[5] Stefan Bruckner and Torsten Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1468–1476, 2010.

[6] Yuchen Karen Chen, Yong S. Chu, JaeMock Yi, Ian McNulty, Qun Shen, Peter Voorhees, and David Dunand. Morphological and topological analysis of coarsened nanoporous gold by x-ray nanotomography. *Applied Physics Letters*, 96(4), 2010.

[7] Ismail Demir, Mihaela Jarema, and Rüdiger Westermann. Visualizing the central tendency of ensembles of shapes. In *Siggraph Asia Symposium on Visualization*, pages 3:1–3:8, 2016.

[8] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. *Computer Graphics Forum*, 35(3):221–230, 2016.

[9] A.G. Gyulassy, M.A. Duchaineau, Vijay Natarajan, V. Pascucci, E.M. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE Trans. on Vis. and Computer Graphics*, 13(6):1432–1439, 2007.

[10] M. C. Heiber. Ising_opv v2.0. `https://github.com/MikeHeiber/Ising\_OPV`.

[11] Michael C. Heiber and Ali Dhinojwala. Efficient generation of model bulk heterojunction morphologies for organic photovoltaic device modeling. *Phys. Rev. Applied*, 2:014008, 2014.

[12] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840, 2002.

[13] Ulrike Homberg, Daniel Baum, Steffen Prohaska, Ute Kalbe, and Karl Witt. Automatic Extraction and Analysis of Realistic Pore Structures from μCT Data for Pore Space Characterization of Graded Soil. In *Proceedings of ICSE-6*, pages 345–352, 2012.

[14] A. Huang, G. M. Nielson, Anshuman Razdan, G. E. Farin, D. P. Baluch, and D. G. Capco. Thin structure segmentation and visualization in three-dimensional biomedical images: a shape-based approach. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):93–102, 2006.

[15] Mark Jones, J Andreas Baerentzen, and Milos Sramek. 3D distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.

[16] Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen, and Torsten Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *IEEE Visualization*, pages 513–520, 2003.

[17] O.D. Lampe, C. Correa, Kwan-Liu Ma, and H. Hauser. Curve-centric volume reformation for comparative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1235–1242, 2009.

[18] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2654–2663, 2014.

[19] Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer, and Ajith Mascarenhas. Robust on-line computation of Reeb graphs: Simplicity and speed. *ACM Transactions on Graphics*, 26(3), 2007.

[20] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova. Hierarchical Stochastic Neighbor Embedding. *Computer Graphics Forum*, 35(3):2579–2605, 2016.

[21] Chris Pudney. Distance-ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. *Computer Vision and Image Understanding*, 72(3):404–413, 1998.

[22] Biswajit Ray and Muhammad A Alam. Random vs regularized OPV: Limits of performance gain of organic bulk heterojunction solar cells by morphology engineering. *Solar Energy Materials and Solar Cells*, 99:204–212, 2012.

[23] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace–Beltrami spectra as 'Shape-DNA'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[24] O. Soldea, G. Elber, and E. Rivlin. Global segmentation and curvature analysis of volumetric data sets using trivariate B-spline functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):265–278, 2006.

[25] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.

[26] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, J. M. Verbavatz, and S. Bergner. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Res.*, 9:2579–2605, 2008.

[28] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, 2013.

[29] O. Wodo and B. Ganapathysubramanian. Modeling morphology evolution during solvent-based fabrication of organic solar cells. *Computational Materials Science*, 55:113–126, 2012.

[30] O. Wodo, S. Tirthapura, S. Chaudhary, and B. Ganapathysubramanian. A graph-based formulation for computational characterization of bulk heterojunction morphology. *Organic Electronics*, 13(6):1105–1113, 2012.

[31] Olga Wodo, Srikanta Tirthapura, Sumit Chaudhary, and Baskar Ganapathysubramanian. Computational characterization of bulk heterojunction nanomorphology. *Journal of Applied Physics*, 112(6), 2012.

[32] Kelin Xia, Xin Feng, Zhan Chen, Yiying Tong, and Guo-Wei Wei. Multiscale geometric modeling of macromolecules I: Cartesian representation. *Journal of Computational Physics*, 257, Part A:912–936, 2014.