A Fractional Cartesian Composition Model for Semi-spatial Comparative Visualization Design

Ivan Kolesár, Stefan Bruckner, Ivan Viola, and Helwig Hauser

Abstract—The study of spatial data ensembles leads to substantial visualization challenges in a variety of applications. In this paper, we present a model for comparative visualization that supports the design of according ensemble visualization solutions by partial automation. We focus on applications, where the user is interested in preserving selected spatial data characteristics of the data as much as possible—even when many ensemble members should be jointly studied using comparative visualization. In our model, we separate the design challenge into a minimal set of user-specified parameters and an optimization component for the automatic configuration of the remaining design variables. We provide an illustrated formal description of our model and exemplify our approach in the context of several application examples from different domains in order to demonstrate its generality within the class of comparative visualization problems for spatial data ensembles.

Index Terms—Visualization Models, Integrating Spatial and Non-Spatial Data Visualization, Design Methodologies

1 Introduction

With the continued improvement of data acquisition technology, both in terms of measurements as well as numerical simulation, we see a new trend towards studies of large and multi-faceted data ensembles. Frequently, such ensemble studies also focus on spatial aspects of the data, for example, cohort studies in medicine that are based on population screening or parametric ensemble studies that are based on numerical simulation in engineering. Such studies are intrinsically challenging, not only because of large data volumes, but in particular also because of the multi-faceted character of the study, including multi-dimensionality, data heterogeneity, multi-modality, etc. [17].

Recently, visualization research has picked up this challenge of supporting the study of entire data ensembles (as compared to the more traditional challenge of visualizing individual datasets) and first results are available also for the specific challenge of visualizing *spatial* data ensembles [4, 19]. With *spatial data*, we denote all those cases, where certain spatial aspects of the data are central to the visualization question, for example, the 3D anatomy of the screened individuals or the 2D flow structures that are simulated by computational fluid dynamics. Naturally, when visualizing ensembles—and in particular, when studying spatial data ensembles—, the *comparative visualization* of ensemble members becomes a central and non-trivial challenge, notably when multi-dimensional ensembles are studied or many ensemble members should be compared.

A central theme in comparative visualization is to enable the user to visually relate two or more ensemble members to each other, often also in a structured form [11], and the study of intra-ensemble variability is also central to many cases of visual parameter space analysis [31]. In this paper, we address the particular challenge of integrating spatial and non-spatial data in the comparative visualization of multi-dimensional ensembles, where the user is interested in preserving selected spatial aspects of the data as much as possible, while still comparing multiple ensemble members in one visualization.

The key question of interest here—and a central question in much of the related work, also—is the following: Given that the user is interested in a spatial visualization of the data, what is an optimal tradeoff between spatial abstraction on the one hand (in order to visualize multiple ensemble members together) and preserving the data spatiality on the other hand (and thus limiting the number of members to compare). See Fig. 1 for an illustration of this abstraction–composition interplay. A well-chosen compromise is obviously needed, since the visualization space—2D or 3D, usually—is limited. Enabling the partial abstraction of data spatiality, for example, through a simple 3D→2D projection, opens up for the semi-spatial comparison of multiple ensemble members in a joint visualization space. The amount of abstraction is clearly a critical parameter in order to optimally exploit the potential benefit of comparative visualization.

In this paper, we now present a formal model for the design of according comparative visualization solutions in the context of spatial data ensembles. Accordingly, we are able to streamline the process of visualization design. Only a minimal set of visualization parameters is chosen by the user—the remaining degrees of freedom are then optimized automatically, based on the new model.

According to Munzner [26], visualization design is done on four layers: domain characterization, data and task abstraction, designing the visual encoding and interaction, and implementation. Each of these stages carries risks and costs. According to Van Wijk [34], it is of greatest importance to limit the initial costs of visualization design as much as possible, especially in the situation of visualization solutions for experts. Consequently, automatic visualization design, or at least computationally-assisted visualization design, is one important challenge in visualization research—already for many years [24, 40, 36, 30].

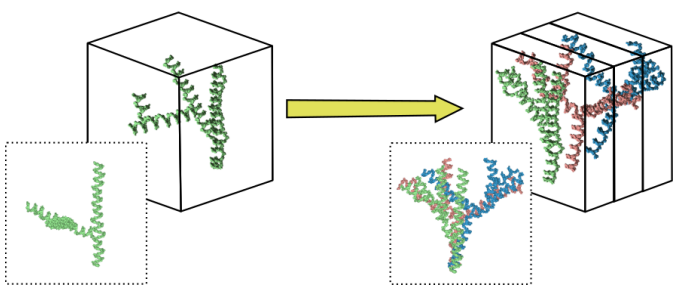

Fig. 1. Partial spatial abstraction—from 3D to slabs (2½D)—enables the composition of a comparative visualization (here of three simulations of a model of polymerization): the more spatial abstraction is done, the more visualization space is available for the comparison (at the cost of a reduced preservation of the spatiality of the data).

• *I. Kolesár, St. Bruckner and H. Hauser are with the Department of Informatics, University of Bergen, Norway, http://www.ii.uib.no/vis/*
• *I. Viola is with TU Wien, Austria, Viola@cg.tuwien.ac.at*

Manuscript received 31 Mar. 2016; accepted 1 Aug. 2016. Date of publication 15 Aug. 2016; date of current version 23 Oct. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2016.2598870

Working out formal models for visualization design helps to enable a structured approach as several recent works document [26, 31, 18, 8]. Thus, our new model is predominantly a contribution to (a) formalizing the tradeoff between spatial abstraction and preservation and (b) enabling an optimization-assisted design of comparative visualization of spatial data ensembles. The formal model also helps with relating visualization solutions to each other, to possibly map them to each other, and to discuss them in terms of costs and benefits.

2 Related Work

In the following, we discuss a selection of related work that addresses *visualization design*, *comparative visualization*, and, in particular, the comparative visualization *based on spatial abstraction*.

2.1 Visualization Models and Design

Visualization design is the process of finding a (good) solution in the space of possible visualizations. Usually, it is focused on a particular problem domain—general-purpose visualization is only reasonable for comparably simple tasks [26]. In our case, we focus on the case of comparative visualization of spatial data ensembles.

In the case of non-spatial data visualization, important pioneering work was done already in the mid-1980s by Mackinlay for relational data [24]. A few years ago, Wills and Wilkinson presented a new system for the automatic design of information visualization [40]. Both works emphasize the importance of the visual expressiveness, and argue for the automatic design of the visualization, if possible. Scientifically acquired knowledge about the effectiveness of certain visual encodings can be used to optimize visualization design, at least partially. An optimal visualization, however, also depends on the user (visualization skills, previous knowledge, as well as the user's task and interest), which is much harder to capture computationally. Therefore, alternative approaches are also pursued that assist the user in the visualization design, instead of fully automating it.

Accordingly, recommendation models for information visualization have been suggested, for example, based on semantic knowledge about the data [36], where the additional information about the data helps in generating a suitable visualization, or based on a probabilistic sampling of a tightly constrained design space [29], helping with the coloring or spatial layout of the visualization. For the case of spatial data visualization, in particular when considering the partial spatial abstraction to compose a comparative visualization, no corresponding solution is available yet.

To compare and evaluate visualization designs quantitatively, formalization is required. In related work we find predominantly two types of formalization: categorical and algebraic. An example of categorical formalization is the conceptual framework of parameter space visualization by Sedlmair et al. [31]. An example for an algebraic formalization is the work on product plots by Wickham and Hofmann [39]. More recently, a similar approach was pursued to address visualization in general by Kindlmann and Scheidegger [18], where a formal mapping between the data and the visualization, between different types of data representation, and between different visualization mappings is described. This work provides a great basis for the analysis and evaluation of visualization in general.

With respect to the question of optimally preserving data properties in the visualization and consequently automating the according visualization design, the recent work by Demiralp et al. [8] is very interesting. They demonstrate, how distance-preserving visualization can be automated, based on crowd-sourced data, and show further, how the notion of visual product spaces assists the estimation of optimal visualization also beyond the acquired data. Also related, we see approaches that assist visualization design by optimizing formal quality measures (mostly in information visualization, however). One such example is the automatic search for an optimal dimension reduction in parallel coordinates by Johansson and Johansson [15]. However, in all related work that we reviewed, we could not find a solution that addressed the (partial) spatial abstraction in order to enable the composition of a comparative visualization, however.

2.2 Comparative Visualization

To compare the individual members of a (spatial) data ensemble, comparative visualization is obviously a natural solution. Most generally, comparative visualization was studied in the context of information visualization by Gleicher et al. [11]. In their comprehensive and thorough survey, three basic layouts of comparative visualization are analyzed and exemplified: *juxtaposition*, *superposition*, and *explicit encoding*. Even though this study is primarily focused on non-spatial data visualization, we still find their categorization general enough to also help with the comparative visualization of spatial data ensembles, and thus incorporate their formalism in our model, as well.

Related to the notion of comparative visualization is also the one of composition views [14]. This design model describes strategies to combine visual representations in the same geometrical space, in particular *juxtaposition*, i.e., placing visualizations side-by-side, *superimposition*, i.e., overlaying visualizations, *overloading*, i.e., utilizing the space of one visualization for another, and *nesting*, i.e., nesting the content of one visualization inside another. The first two categories are similar to the one by Gleicher et al. [11] and we adopted them in our model. The latter two, however, seem less suitable for our type of comparative visualization, in particular, when considering a larger number of ensemble members for one visualization.

2.3 Comparative Visualization using Spatial Abstraction

Closely related to our work, a few interesting studies have been published recently, based on both comparative visualization as well as on spatial abstraction. Klemm et al. [19] demonstrated visual analytics of rich epidemiological data related to back problems from a large cohort study in Germany. Krekel et al. [22] used spatial abstraction and superposition to study limb movements with comparative visualization. Waser et al. [38] present an advanced system that enables the visual analysis of rich simulation ensembles, helping, for example, with the planning of how to meet different types of flooding scenarios. Alabi et al. [1] presented a comparative visualization of surface ensembles, where a limited number of surfaces are cut into stripes and put onto each other. While spatial visualization is central to these examples of comparative ensemble visualization, and we also think that it would be meaningful to analyze them in terms of our model, the notion of spatial abstraction is still not addressed as explicitly as in our work.

Closest related to our work, are solutions which focus on the (partial) spatial abstraction of ensemble members for an effective comparative visualization. Selected examples can be found in medical visualization, for example, for the comparative visualization of arterial blood flow [4], where the human aorta is virtually straightened to enable an effective side-by-side visualization. Through a careful (partial) spatial abstraction, the traditional 3D frame of human anatomy is transformed into the comparative visualization space, where then one axis corresponds to the length of the aorta and the other axis is "freed up" for the comparative visualization of the blood flow at different time steps.

Also in medical visualization, Termeer et al. [33] have transformed the shape of the heart from 3D to 2D, encoding the coronary arteries and their respective territories. The deformation enables the visual encoding of the spatial blood supply information, and also the comparative visualization of blood supply from the individual coronaries. Slices are stacked on top of each other, creating a volumetric bulls eye visualization [32], helping to understand the spatial correlation between the supply of different coronaries in different sections of the heart. Similarly, Köhler et al. [20] have abstracted the aortic flow to a circular plot to compare the blood flow between different patients. Yet another example is curve planar reformation [37, 16], where the central curves of anatomical structures are aligned to enable the comparison of different structures. Also related, Eikeland [9] combines different types of spatial abstraction for multi-volume visualization.

The idea of using spatial deformations to facilitate an effective comparative visualization was also used for flow visualization [35, 13] and the visualization of geoscientific data [23]. For volumetric data, Busking et al. [6] have defined deformations for direct focus+context visualization, while (or even though) showing only one dataset at a time.

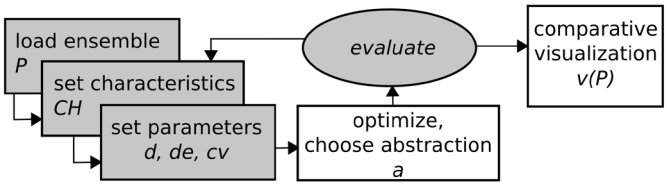

Fig. 2. Optimization-assisted design of comparative visualization—the user defines, which spatial data characteristics are important (CH) and sets central parameters: the dimensionality of the visualization (d), number of ensemble parameters to explore (de), and the class of comparative visualization to choose from (cv); an automatic optimization step completes the visualization design, which then is evaluated by the user.

All these specific examples demonstrate that the design of comparative visualization for spatial data ensembles is a general challenge that spans a variety of application domains. A formal model that can lead to a streamlined design process is therefore considered a potentially valuable contribution.

3 The Model

In order to introduce our model, we first specify our context, i.e., the comparative visualization of spatial data ensembles, and illustrate this setting with a synthetic example in order to support presentation of the formal explanation of our model.

Focus: We assume that a user is interested in some phenomenon or artifact that is available to the user in form of an ensemble dataset. We do not see any major restriction with respect to the type of ensemble data and address both measured ensembles as well as computed ensembles, for example, from cohort studies based on population screening or from numerical simulation. Such an ensemble could be a parametric ensemble, for instance, based on the structured variation of simulation parameters, or a stochastic ensemble, generated by the repeated simulation of a stochastic model. Other types of ensembles are possible as well. We also assume that the user is interested in certain non-trivial spatial characteristics of the phenomenon/artifact, for example the shape of the ensemble members—if this would not be the case (as in many cases of information visualization, for example), the complete abstraction of all data spatiality likely would lead to an effective visualization, for example, in a scatterplot. Accordingly, we focus on use cases, where the user is interested in preserving as much data spatiality as possible, while still composing an effective comparative visualization and enables the swift and flexible study of rich spatial data ensembles.

Illustrative example: To complement the formal description of our model, we introduce a hypothetical example as an illustration: Imagine a user with interest in ivy-covered walls and assume that an ensemble dataset is available, which contains a two-parameter ensemble of simulated, ivy-covered walls, where one parameter encodes the ivy's likelihood to branch, while the other parameter encodes the ivy's age (time of growth). The user judges ivy-covered walls in terms of certain spatial characteristics, including left–right symmetry, the overall coverage of the wall, and the number of wall spots, which are not covered by ivy. The ensemble contains 10×10 spatially-detailed ivy-models so that some spatial abstraction is needed to compose an informative comparative visualization on the screen. The very left of Fig. 5 and the top row of Fig. 6 show selected examples from this ensemble.

With our work, we address the situation, where partial automation is preferred over completely automatic optimization—we assume that the user has a few preferences that are provided as input to the design process. We assume, however, that the user is not interested in technical particularities of the visualization design, relating to those parts of the process, where optimization is exploited, accordingly. In the following, we first explain how the user is part of our model, before we then explain our model and the according visualization design process.

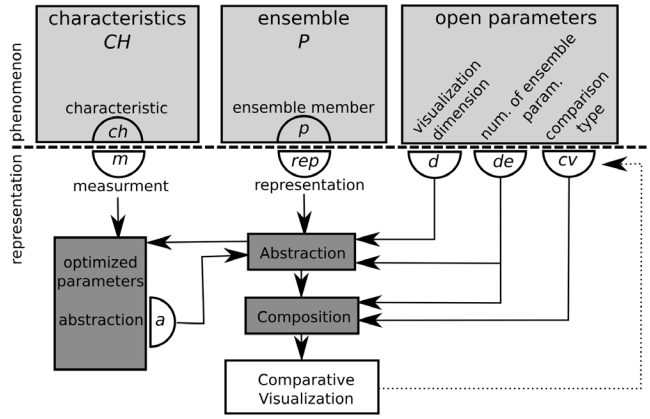

Fig. 3. Model overview—above the dashed line, we show the user-side of our model (a certain ensemble P is considered, together with selected spatial characteristics CH, as well as user-chosen parameters) and relate it to the formal model: an ensemble member p is represented (rep) and a characteristic ch is measured by a corresponding measurement function (m). The visualization is composed of (partially) abstracted member representations, assisted by an optimization component, choosing the optimal abstraction.

3.1 The User's Role

In order to capture the entire design process, we keep the user centrally in the picture—after all, it's the user, who initiates the process, based on data and a task, before then steering it to a satisfying visualization.

We assume that the ensemble dataset P is given as an n-parameter set of ensemble members p_i, which are represented in the data in a certain form $rep(p_i)$, for example as a geometric model or as image data. In the synthetic ivy-example, n is 2 (branching-likelihood & ivy-age) and $rep(p_i)$ is a geometric tree-model with branches and leaves.

Based on our exchange with collaborators from different domains, we assume that the user wishes to choose which type of comparative visualization is designed (a juxtaposition, superposition, or a combination of them)—accordingly, we enable the user to choose this type-parameter cv as one of $\{juxtaposition, superposition, hybrid\}$. We note that the choice of cv also could be optimized, given that the user is agnostic concerning the type of the comparative visualization.

Depending on the complexity of ensemble P (in terms of its dimensionality n), more or less complex visualization designs are possible. The user could prefer to examine the intra-ensemble variability with respect to individual ensemble-parameters, for example, one by one. Alternatively, the user could be interested in a visualization setup that compares the ensemble members with respect to several parameters at once. We see this question closely related to the user's task and thus enable the user to provide this input as another parameter, de, i.e., the number of ensemble parameters to be varied.

Thirdly, we also leave it to the user to choose the target dimensionality of the visualization, i.e., 2D or 3D (d). In principle, it can be argued that 3D visualization offers more space for better accommodating ensemble members, in particular when many are to be compared. 3D visualization, however, comes at the cost of visual comparisons that are more difficult to interact with and to interpret. It seems non-trivial (while not impossible) to automatically measure this additional cost and thus enable an automatic optimization of this parameter, also.

In our illustrative example, the user could experiment with a few, semantically meaningful settings that lead to different comparative visualization solutions. Choosing a *juxtaposition* (cv) of ivy time-sequences (de) in 2D (d), for example, would require a certain spatial abstraction to enable the accommodation of multiple ensemble members side-by-side, whereas the same specification in 3D (d) would lead to a composition of non-distorted ensemble members (behind each other) at the cost of a more difficult interaction and interpretation.

3.2 A Cartesian Composition Model

Any type of comparative visualization (juxtaposition, superposition, or a combination) competes about a joint resource in terms of the visualization space in which it is constructed, bringing together multiple ensemble members. Overlaying them in the same space (by superposition) leads to a competition about color and opacity. Using semi-transparent representations, for example, allows to visually compare a certain number of them. The scalability of superposition, however, is limited and more than one ensemble parameter rarely ever be successfully varied through this mechanism. The side-by-side arrangement of ensemble members for the comparative visualization of intra-ensemble variability, on the other hand, leads to a competition about the joint visualization space itself (both in terms of extent and dimensionality)—the more ensemble members come together, the more they get confined in terms of the space that they can claim. To fit them into accordingly confined parts of the visualization space, spatial abstraction mechanisms are used, including simple orthographic projections (for example from 3D to 2D) or more advanced deformation procedures (for example the straightening of certain structures).

Depending on the number of ensemble members to be shown (per ensemble parameter to be varied)—10 in the ivy-example for both branching-likelihood and ivy-age—a discrete (for small numbers) or near-continuous decomposition of the visualization space is reasonable, leading to a potentially fractional decomposition of the visualization space dimensionality: The 2D visualization space could be decomposed, for example, into a juxtaposed set of narrow stripes (for a low-number, one-parameter study), into a 2D grid of patches (for a two-parameter study—both variations of low cardinality), or into a dense set of lines (for a high-number, one-parameter study).

The number of all possible decomposition types is limited and well organized—see Fig. 4 for an illustration of all possible decomposition types for a 3D visualization. Which decomposition actually leads to the best result for the user, is difficult to predict automatically, and we assume that the user provides this choice as a parameter.

In the following, we now describe the entire model formally (as a basis for then describing the actual design process). An graphical overview of our framework is provided as Fig. 3.

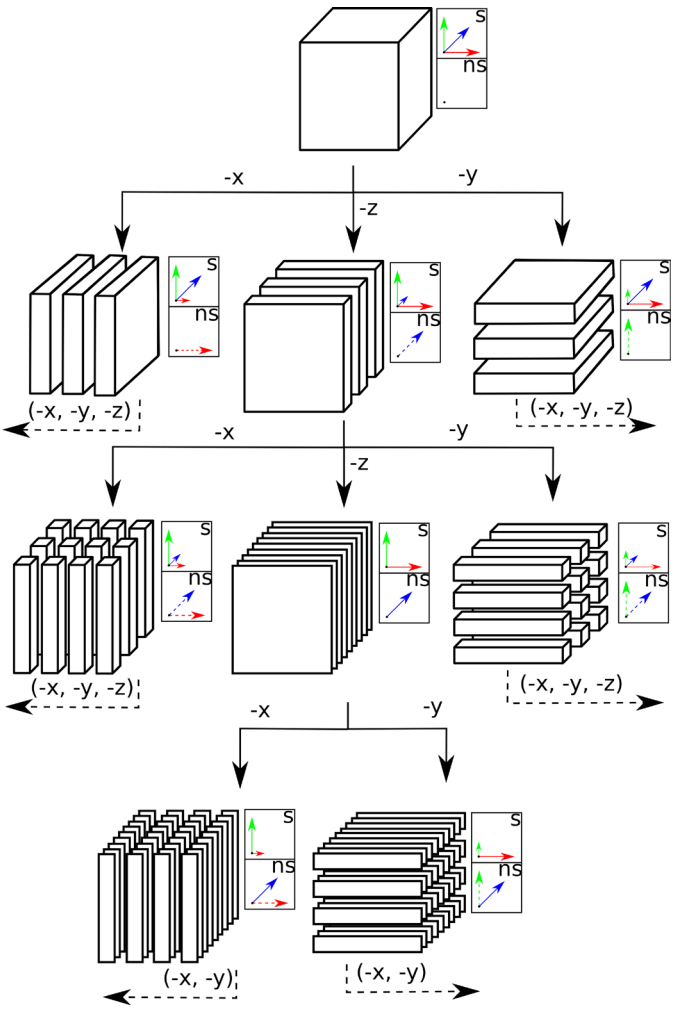

Fig. 4. Overview of decomposition opportunities for a 3D visualization space, where parts are devoted to spatial data aspects (s), while others (ns) are devoted to non-spatial aspects. Each dimension can be decomposed discretely (in the case of few items) or (near-)continuously (for many), amounting to a fractional, Cartesian composition model for comparative visualization based on juxtaposition.

3.3 Formal Model Description

In abstract terms, we define a comparative visualization $v(P)$ of a spatial data ensemble P as a visual composition cv of (partially) abstracted spatial representations $a(rep(p))$ of the ensemble members p that should be compared:

$$v(P) = cv(\{a(rep(p)), p \in P\}) \tag{1}$$

Abstraction procedure $a(.)$ takes the representation of ensemble members $rep(p)$ as input and transforms it so that $a(rep(p))$ is of an extent and a dimensionality which is suitable for composition through function $cv(.)$, leading to the desired comparative visualization. To generate the top row of Fig. 6, for example, function $a(.)$ would amount to a simple, isotropic scaling, and function $cv(.)$ would be juxtaposition.

We have worked out our model for three types of composition (cv):

- Juxtaposition: side-by-side comparison.
- Superposition: members share the same space.
- Hybrid: a combination of juxtaposition and superposition, for example side-by-side comparison of element stacks.

According to Gleicher et al. [11], also the explicit encoding of differences between ensemble members may be supported. Even though we have not implemented this option in our framework, we envision that integrating explicit coding as another option for cv should be doable.

Juxtaposition places the objects next to each other in the available visual space. This approach relies on the viewer's memory to visualize the differences and similarities between objects. Such an approach is well-suited to see patterns between visual objects and to explore relations among multiple data dimensions [14]. Comparing distant parts of the visualization, however, is challenging in such a comparison.

Superposition places two or more objects on top of each other in the same space. This approach is used to highlight spatial relations between the compared objects, but suffers substantially, when a larger number of objects are to be compared. The simplest solution, i.e., making the images semi-transparent has also issues with cluttering and scaling beyond only few objects to compare [11]. To overcome this problem, additional visualization encodings are required, for example coloring of different ensemble members (see Fig. 5).

The *combination of juxtaposition and superposition* is common in creating comparisons [11], for example, superposition in one dimension and juxtaposition in all others. Also this approach should be used with care. On the one hand, it enables to explore patterns and spatial relations between multiple members. On the other hand, it is more complex and can be challenging to read.

Other types of composition (of 2D representations) can lead to yet different visualization designs. Stacking explicit encodings of differences, for example, can lead to a comparative visualization in the style of certain video visualization solutions [7] or other existing visualization design [41].

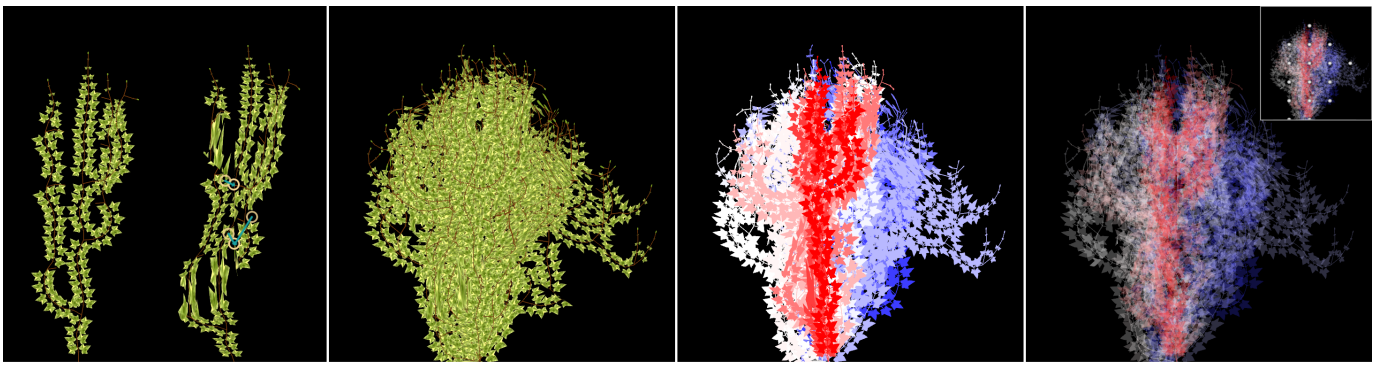

Fig. 5. Example for a particular abstraction $a(.)$ that supports the visual study of a particular spatial characteristic (here holes in the ivy-coverage) through superposition. The original structure (very left) is deformed such that "holes" in the ivy-coverage are snapped to a given grid (orange rings). Three options for a comparative visualization are shown to the right: one without any additional visual encoding (hard to appreciate), one where color differentiates ensemble members, and one with additional transparency (on the very right), where the holes can be interpreted.

A key ingredient to our comparative visualization model is the abstraction function $a(.)$ and the optimal choice of this function has a substantial influence on the effectiveness of the resulting visualization. In our model, we suggest to choose this abstraction function a from a set of available functions A (A could be discrete or continuous) in order to minimize a residual function (error measurement $e_{a,ch}$) that evaluates, how well a particular abstraction a is capable of preserving the spatial characteristics of interest ch as selected by the user:

$$a : \underset{a \in A}{arg\,min}\; e_{a,ch} \tag{2}$$

Given a suitable residual function (described below), we can use optimization in this step to automatically select an abstraction a that preserves the data spatiality of interest best, while still confining the ensemble member representations to the available space in the comparative visualization. In order to compute to which degree a certain abstraction does preserve spatial characteristics of interest best-possibly, we refer to measurement functions $m(.)$ that provide a quantitative evaluation as illustrated also in Fig. 3, both for the original representation $rep(p)$ as well as for the abstracted representation $a(rep(p))$:

$$e_{a,ch} = m_{ch}(rep(p)) - m_{ch}(a(rep(p))) \tag{3}$$

Measurement function $m_{ch}(.)$ results in a quantitative evaluation of spatial characteristic ch that enables to assess to which degree an abstraction preserves the respective data characterstic. In the ivy-case, for example, we measure the spatial characteristic "wall coverage" by computing the fractal dimension of the ivy tree—the closer to 2, the better the coverage. Another example would be the *mean area exponent* for 2D tree structures as defined by McGuffin and Robet [25] in order to measure space-efficiency in hierarchical data visualization.

Reducing the dimensionality and/or the extent of member representations usually leads to a measurable reduction in terms of m_{ch}-values—if $a(.)$ is the identity map, however, $e_{a,ch}$ equals 0, of course. Projecting a 3D ivy model into 2D, for example, would compromise the thickness of the ivy-cover, but not the areal coverage across the wall. A more radical abstraction will usually correspond to a larger reduction in terms of m_{ch}-values, meaning that we can select the best-possible abstraction by minimizing $e_{a,ch}$.

By abstraction, in general, we mean a process that confines the spatial representation in terms of dimensionality and/or extent in order to make visualization space available for the composition of the comparative visualization. The available abstraction functions play a critical role in our model. We identified at least four types:

- *Projection*: simple orthographic projections, for example from 3D to 2D, are straight-forward and clearly useful for dimension reduction—projections are comparably easy to interpret and preserve certain spatial characteristics without special solutions.
- *Global deformations, usually simple*: global transformations (for example isotropic scaling) that are usually straight-forward to calculate—abstractions of this type are useful for confining the extent of representations, while still being well-behaved and esay to interpret (see also the survey by Gain and Bechmann [10]).
- *Piece-wise deformations, often more complex*: advanced abstraction mechanisms that enable a semantically meaningful, but often also complex transformation, for example, through straightening and/or flattening approaches [3, 23].
- *Combinations*: additionally, approaches from the above mentioned classes can also be combined, for example, through the iterative application of several of them—an example would be an orthographic projection, followed by isotropic scaling (Fig. 4).

Providing an appropriate set of available abstractions is crucial for the potential success of an optimization based on this model, as well as the opportunity to formulate quantitative measurement functions to capture the spatial characteristics of interest. Further, it is common that not only one spatial characteristic is of interest for the user. In this case, the residual function e (to be minimized) can be formulated as a weighted sum of the residual functions per characteristic:

$$e = \sum w_{ch}\, e_{a,ch} \tag{4}$$

The weights w_{ch} can be used to balance the individual per-characteristic residuals $e_{a,ch}$, before the optimization.

3.4 Model-based Visualization Design

For one instance of designing a comparative visualization according to the above described model, we assume that the spatial ensemble dataset is given (encompassing the spatial representations of the ensemble members, for example, in the form of a mesh, image, or volume), as well as the corresponding user interests (in terms of spatial characteristics and the according measurement functions).

As a first step, reference values are computed (for the efficient computation of residuals, subsequently) by applying the chosen measurement functions to the given data ensemble (no abstraction). It is sufficient to evaluate the measurement function on a representative selection of ensemble members. This becomes relevant, if the ensemble is large and/or if the measurement functions are costly to evaluate.

The next step is that the user chooses the type of comparative visualization cv to use and the parameters de that should be varied. This leads to a set of possible layout candidates for the comparative visualization (a subset of what is illustrated in Fig. 4).

To accommodate the ensemble members in the comparative visualization according to the chosen composition, an appropriate abstraction is needed per ensemble member. For abstraction functions $a \in A$,

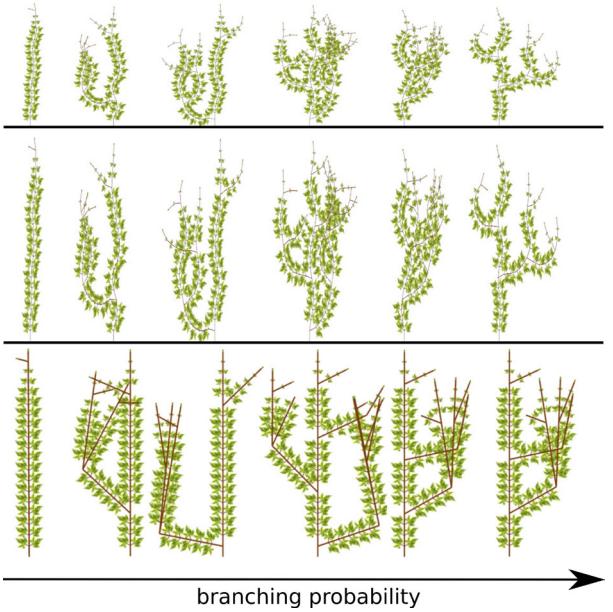

Fig. 6. Three examples of comparative visualization, based on different abstractions (1st row: isotropic scaling; 2nd row: anisotropic scaling)—while the representations are fitted to the available space, the branching structure is difficult to compare. An alternative abstraction is used in the 3rd row that optimizes the preservation of the length of branches and the overall branching structure.

the residual $e_{a,ch}$, i.e., the evaluation of the corresponding measurement function minus the previously computed reference value, is computed (or e in the case of multiple characteristics of interest). This drives the optimization in order to choose one a with minial residual. Depending on the size of A, this optimization can be very simple (evaluate the residual for all $a \in A$ and choose the one with the smallest residual). In the case of a larger set A (for example, when considering parameterized abstraction functions) or even a continuous set A, an appropriate iteration-based minimization algorithm (for example, a conjugate-gradient-based method) can be used.

Eventually, the ensemble members are abstracted to $a(rep(p))$ and composed into the comparative visualization according to cv. The user evaluates the visualization $v(P)$ then and iterates, changing parameters, i.e., d, cv, or de, or concludes, if satisfied (Fig. 2).

4 Demonstration Examples

To demonstrate the power and utility of our model, we show three illustrative examples. First, we further detail our ivy example as an instance of a 2D dataset with two ensemble parameters (Sect. 4.1). Our focus here is to show the iterative analysis process and to demonstrate the preservation of selected spatial characteristics. The second example demonstrates the power of spatial reduction for the comparative visualization of 3D data (Sect. 4.2). The third example relates to the domain of city-planning, for example, for game design: we show a hybrid composition solution for exploring a 3D ensemble (Sect. 4.3). For each example, we discuss the spatial characteristics of interest, i.e., the features that should be preserved under abstraction and their corresponding measurement functions (the quantitative measure used to capture the characteristic), as well as the chosen abstraction and then depict and discuss the resulting visualizations.

4.1 Wall-covering Ivy

We synthesized an ensemble as an open L-System [27], simulating the effect of varied parameters (environmental, genetic, ...) on the growth of ivy on a planar wall. We have modeled the branching probability, nutrition flow, and segment length, as well as the available space in the environment and the light level as parameters. As an initial simple example, we focus on a subset of this ensemble, which consists of ivy plants growing over time with different branching probabilities. Our goal is to study the shape variability in the ensemble, in particular with respect to the differences in branching probability. In total, the visualized ensemble subset has 10×10 members.

The individual ivy instances p of the ensemble P_{ivy} are represented as meshes, with an explicit encoding of the tree structure g. To exemplify the core principle of our model, we consider three spatial characteristics, together with the corresponding measurement functions:

1. *coverage*, indicating the fraction of space covered by the plant, with the measurement function m_{cov} that estimates the fractal dimension of the plant by the box-counting method [2].
2. *numbers of holes*, capturing how "complete" the coverage is, with the measurement function m_{noh}, counting the number of connected of closed regions inside the ivy structure.
3. *symmetry*, characterizing how "balanced" the plant is, with function m_{sym}, quantifying the difference in coverage between the left and right branches, computed in a hierarchical manner.

To demonstrate the selection of an optimal abstraction, we provided a small set of comparably simple abstractions, which have distinct behavior with respect to the preservation of spatial characteristics:

1. *1D-scale*, i.e., a straight-forward scaling of one spatial dimension (x or y), denoted as a_{scX}, a_{scY}, leaving most of the geometrical characteristics intact. Other characteristics, including branch lengths, for example, are not preserved.
2. *squeeze*, a_{squ}, a more complex abstraction that focuses on the properties of the branches, preserving the length of the branches, the branching hierarchy, and the coverage. Branches are straightened and rotated towards the main stem, minimizing overlap. This abstraction is inspired by reformation techniques for blood vessel visualization [16].
3. *hole-snap*, a_{hsn}, aims at supporting the comparison of holes. Here, we overlay a coarse grid over the ivy mesh and move all holes to coincide with the closest grid vertex. The results of this abstraction are shown in Fig. 5.

Demonstration: We show results for different parameters and sets of characteristics, starting with a case where the aim is to preserve coverage and branch length, using the model parameters $cv = juxtaposition$ and $de = 1$. Based on these parameters, the ensemble members are put side-by-side. To fit the individual ensemble members into the available screen space, their spatial extent must be confined and abstraction a_{squ} is automatically chosen (see Fig. 6, bottom row), since it has a smaller error for branching and coverage when compared to scaling. As shown in the figure, we get a clear depiction of all individual substructures, including leaves and segments. If, on the other hand, our focus is on the preservation of holes, the resulting composition presents us with a side-by-side view using scaling as abstraction (see Fig. 6, middle row), because the squeeze abstraction changes the topology of branches and therefore affects the number of holes. For comparison, the top-most row of Fig. 6 shows the result if only uniform scaling is permitted.

Interestingly, if we aim to preserve coverage and set the ensemble parameter dimension to two, the proposed composition is also a side-by-side layout, using scaling as abstraction. This is also demonstrated in the supplemental video. If a superposition or a hybrid type of comparative visualization is desired, a_{hsn} is chosen, as it delivers the best results in this case and there is no need for reducing the dimensionality as shown in Fig. 5.

4.2 PARP Polymerization

Our second illustrative example demonstrates how our model supports dimension reduction. We visualize a dataset from the interactive simulation of emergence in polymerization [21]. In this earlier work, we

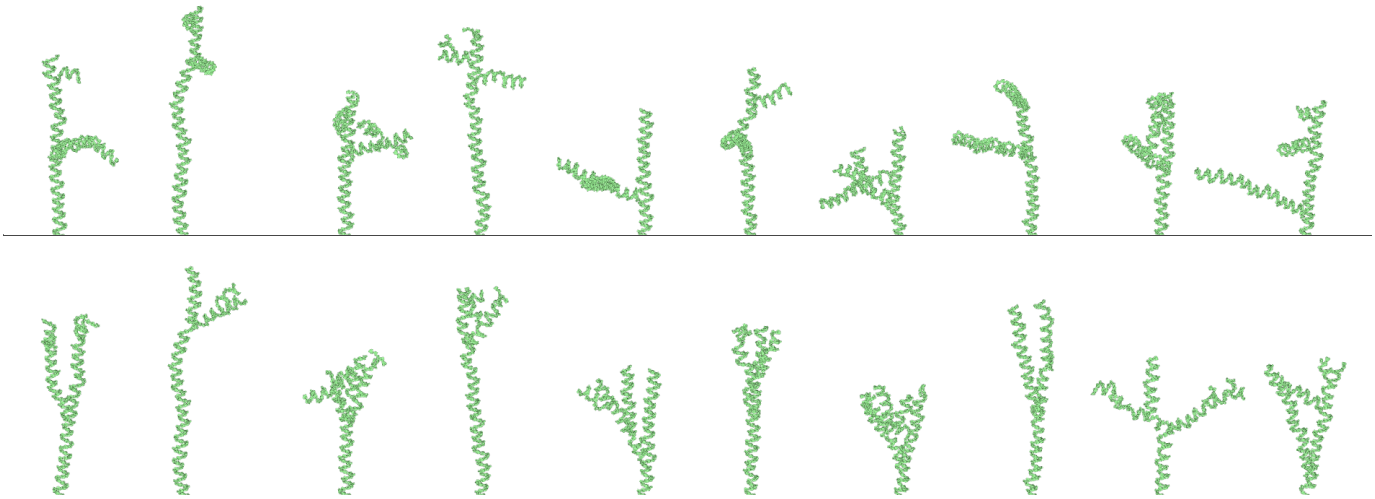

Fig. 7. Two examples of PARP compositions: a default solution (isotropic scaling, top row), a solution based on an abstraction that carefully bends the polymer into a narrow slab (bottom row). While the geometry of the molecule is distorted a bit in the lower example, certain spatial properties, including the length of the branches and the overall branching structure, are preserved in a better way.

have presented an illustrative modeling approach to generate interactive simulations of polymer growth inside an environment filled with molecules and monomers, i.e., the basic building blocks of polymers. Polymers grow as either linear or branched structures and they are represented as strings of symbols, representing the monomers and their respective parameters. The process of polymerization varies based on the parameters of the simulation such as the concentration of the molecules, the branching probability of the polymer, and the duration of the polymerization process. In our dataset we have 41 polymers created with different branching probabilities. The polymers are represented as meshes, augmented by a tree structure which encodes the relationships between the monomers.

The only variable that is changed in our ensemble is the branching probability, which is linearly increasing from 5% to 50%, determining the probability of a newly connecting monomer to form a new branch. The properties of polymers, as for proteins in general, are highly dependent on their structure. It is therefore relevant to explore their topological and geometrical properties. For the purpose of our demonstration, we use a set of simple properties which influences the elasticity of the molecule. The characteristics of interest are

1. The average angle between neighboring monomers, m_{AA}.
2. The maximum length of the structure m_{MLen}, i.e., the maximum distance between the start monomer and all end monomers.
3. The space filling factor m_{SF}, computed as a ratio between the volume of the structure bounding box and the accumulated volumes of the monomer bounding boxes.

We consider the following abstractions:

1. a_{scale} is the same as in the ivy example (in 3D—it is either uniform for all 3 axes, non-uniform for each axis, or a the combination of the two).
2. a_{slab} is a deformation which only rotates the branches in order to put the structure into one narrow slab, similar to ACPR [16], with the difference, that we are not working with the vessel segments, but polymer branch segments. This abstraction is an example for reducing 3D structures into slabs (thick slices). This can be helpful if the secondary structures of a polymers, e.g., the alpha helices, should be preserved.

Demonstration: Since this ensemble is a set with one dimension to explore, possible types of comparative visualization are juxtaposition or superposition. Using juxtaposition (in 2D) will put the structures side-by-side and thereby reduce two dimensions. Superposition, on the other hand, will stack the structures, reducing only one dimension. Using our method the user can more clearly explore the branching and space-filling characteristics. This example also shows the power of combinations in the abstraction mechanism. In Fig. 7, we can see that the avilable visualization space is insufficient for placing 10 polymer structures side-by-side. This can be solved, by default, with a trivial, isotropic scale (Fig. 7, top row), but the visibility of branch lengths and the overall branching hierarchy is compromised in this case. Alternatively, a_{slab}, where all branches are rotated into one slab, shows the structure of polymer branches, their lengths, and also preserves the hierarchy in a better way (bottom row).

4.3 Parameterized Cities

As a third example, we study a three-dimensional ensemble with two spatial dimensions (variations of a modeled 2D city layout) to demonstrate the simultaneous exploration of multiple ensemble dimensions. The dataset is an ensemble of city street networks with model parameters that specify the density of the city population and its coverage. The networks are generated by the method by Parish and Müller [28]. This method comes with several parameters such as the overall pop-

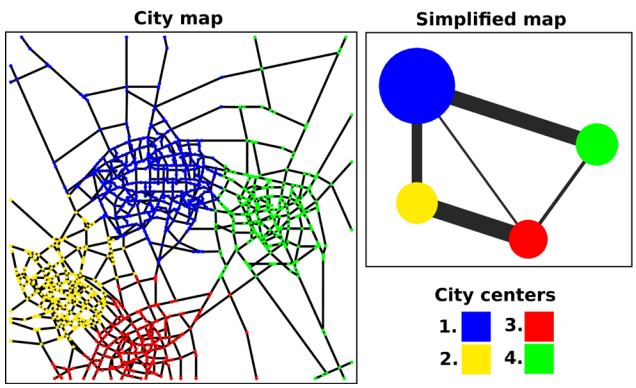

Fig. 8. Illustration of the abstraction used in Fig. 9. Given a certain street network (left), we identify the centers in the network (indicated by color) and represent them as disks—sized according to how many street crossing (colored on the left) belong to the center. Further, we count all street connections between centers and encode this in the width of the edges between the disks (right).

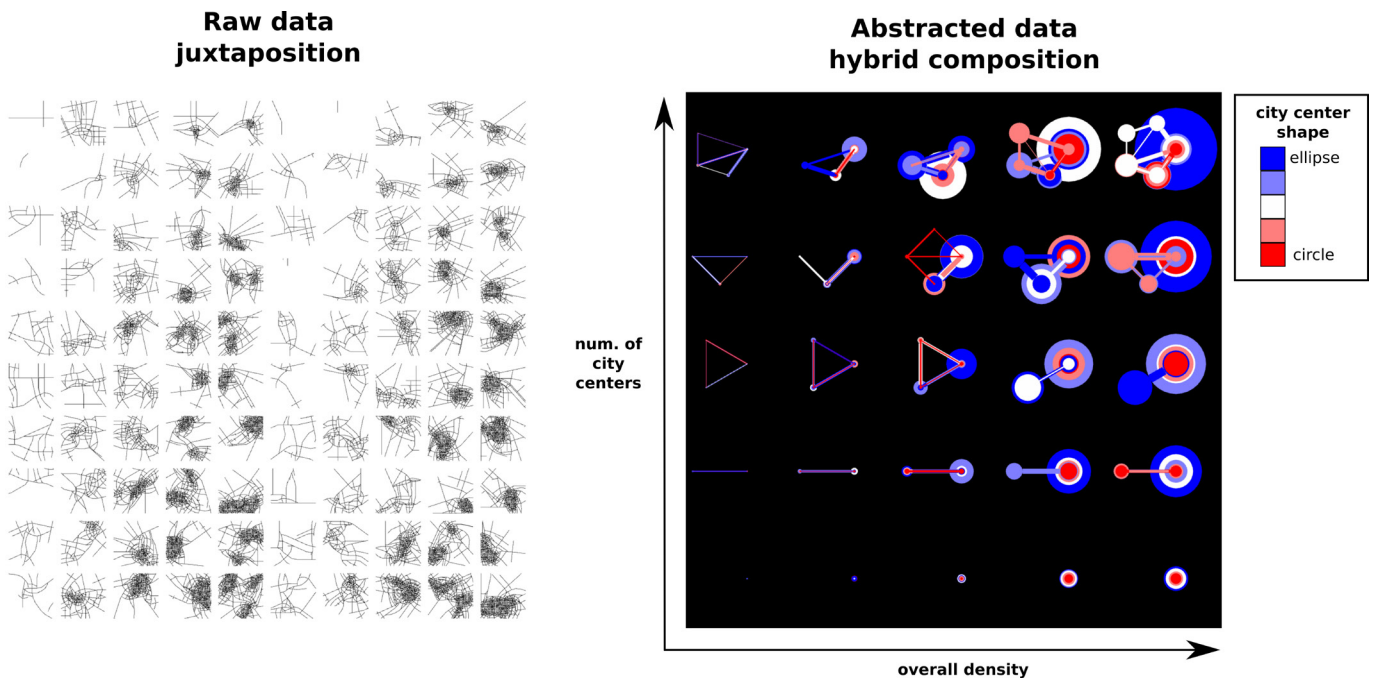

Fig. 9. An optimized variant of a comparative visualization (right) versus a default solution (left). On the left, we see a partial unrolling of the 3D ensemble (not all of the 125 networks in the ensemble are shown). On the right, a hybrid composition is shown with 5×5 juxtaposed superpositions of 5 colored and abstracted networks per patch. The abstraction identifies the centers of the network (represented as disk, sized according to the city center size) as well as their connections (edge width according to the number of street connections between the centers). See also Fig. 8.

ulation density, the water level, height level, and 2D images specifying map properties. We implemented three parameters to generate maps of population density. These parameters are the overall density, the count of the highly-populated areas, and the shape of these areas, which range from circles to ellipses.

Demonstration: In this example we deal with a higher number of ensemble dimensions (three in our case). A comparative visualization can be accomplished using a hybrid solution of positioning stacks of abstracted representations side-by-side in two dimensions (see Fig. 9). In this visualization (Fig. 9, right), we see interesting spatial aspects in this ensemble: while it is not surprising that city centers grow with increasing population density (left-to-right), it is interesting that city centers seem to merge earlier, if they are either circular (red) or very elliptcal (blue). This could be interesting for a city planer, if aiming at relatively highly populated regions with connected, but separated centers (to model non-merging city centers seem to be a challenging task with this city model).

5 Prototype Implementation

To test our model and to work out the illustrative examples from above, we have created a prototype, which is implemented in the Unity3D game engine [12]. The project file is zipped and available online from *http://www.ii.uib.no/vis/projects/physioillustration/research/comparative-visualization-of-spatial-ensemble-data.html*. Even though Unity3D is primarily intended for the development of computer games, its simple C# programming interface provides fast prototyping possibilities and its efficient multi-platform build system supports the sharing of results.

Our visualization exploits 2D and 3D features, as well as the component system of Unity3D. Each ensemble member is represented as a GameObject, which is assigned a component for its visual representation (usually a mesh, but possibly also a network, image, or even a volume). In our demonstration cases, the ivy and PARP examples are represented as meshes, while the city example employed a network representation for the streets. The number of GameObjects in our system is in the order of hundreds.

The measurement functions are implemented as utility functions with GameObjects as input and a floating point number as output. The abstraction functions are static classes with one public function, which takes a GameObject representing an ensemble member as their only parameter. The function returns a new GameObject, which is

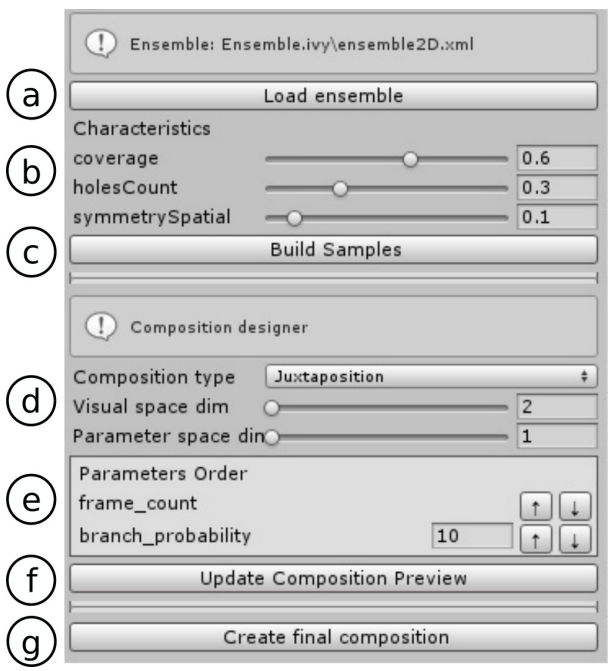

Fig. 10. Screenshot (or screen capture) from the prototype implementation featuring a part of the user interface. User is able to load the data (a, c), change characteristics (b) and model parameters (d, e), and iterate through possible visualizations (f, g).

the abstracted representation of the input ensemble member. These functions only modify the visual representation (e.g., the mesh or network), which they can freely modify (e.g., they may change the type or dimensionality of the representation). For the ease of implementation, all of the algorithms that are used for measurements and abstractions were implemented on the CPU in our current prototype, resulting in a non-optimized performance. Certainly, performance improvements are expected when exploiting the GPU, instead (fully supported in Unity3D).

In our prototype we have used the Unity3D editor interface to interact with the model parameters (see Fig. 10). The first step of the interaction with the model is to load an ensemble (Fig. 10a)—then the characteristics for the example are shown (Fig. 10b). Here, the user adjusts the importance of the characteristics for the optimization (weights w_{ch}). After pressing "Build Samples" (Fig. 10c), our prototype picks ensemble members (randomly), applies the available abstractions and computes the selected characteristics in order to approximate the cost function for the optimization. For all the examples in this paper, we used an empirically-determined sample size of 10 ensemble members. Clearly, this is only a prototype implementation of the otherwise more general framework and ample opportunities for optimization are given here.

After the initial setup, the user interacts with the model parameters, characteristics, and axes (see Fig. 10d, b, e, respectively). After each parameter change, a preview of how the ensemble members will be positioned and how much visual space is available for each member is shown. If the user is satisfied, he/she can then create the final visualization (Fig. 10g).

6 Discussion

Using our model for the design of comparative visualization solutions for spatial data ensembles has several benefits. Firstly, the steered confinement of spatial representations allows to fit more members of an ensemble without loosing important spatial information. This is possible due to the ability of our model to find the abstraction, which preserves the data spatiality of interest best-possibly, i.e., which minimizes the abstraction error. Another important benefit of our model is that it automatically links the choice of the abstraction with the specification of relevant spatiality in the data. Thereby, the time required to explore possibly useful abstractions is reduced. Another benefit of our model is the possibility to swiftly explore the space of different comparative visualization solutions. With the simple change of a parameter, the user is able to see new possible visualizations, based on juxtaposition, superposition, or a hybrid layout. Moreover, the user is able to steer how many ensemble parameters should be varied in the comparative visualization. At large, the overall advantage of our model is the reduction of the overall time that is needed for designing an effective comparative visualization for spatial ensemble data.

As a natural limitation, the overall available space is limited and usually not all the data can be accommodated without any abstraction (which would be the ideal solution, in principle). Our model exploits the fact, however, that not all of the spatial properties are equally important. Therefore, the use of our model is limited, if the user wants to explore all spatial properties at the same time. Specifying the subset of preserved spatial properties in advance also limits the possibility of generating unexpected visualization results and to thereby explore unknown patterns in the non-preserved properties. For such a scenario, a solution based on no abstraction at all would be best, of course.

We have conducted an informal evaluation, based on our model realization for comparatively visualizing polymerization, together with a professor in molecular biology, who is an expert on PARP polymers. We explained our idea, introduced him, based on our prototype implementation, to our process of iteratively creating different instances of comparative visualization (Fig. 2), and we also presented some of our results (in particular, the PARP polymerization as discussed in Section 4.2). He acknowledged that our process design was effective in terms of focusing the interface on questions that really matter for the user (the aspects of the ensemble members that the visualization should bring out, as well as the few open parameters including 2D vs. 3D visualization and super- vs. juxtaposition), while at the same time hiding algorithmic details (handled by the optimization component). He also acknowledged that there is an increased need for comparative visualization due to the emergence of ensemble datasets in many different application cases. With respect to our case study in polymerization, he confirmed that he would prefer the lower visualization in Fig. 7, if his focus was on the number of branches, while he would prefer the upper visualization (in Fig. 7, also), if he would wish to understand, where the density is higher or lower. Beyond the case study, which we had prepared for the discussion, he immediately started to think ahead and suggested new opportunities, for example for the comparative visualization of non-spatial data. In conclusion, he emphasized that he strongly appreciated that he could swiftly survey a larger set of different comparative visualization results (for the same data ensemble) by just adjusting the weights of the different interest functions (Fig. 10b) and by having a new comparative visualization result being computed from these adjustments automatically.

7 Conclusion & Future Work

In this paper, we present a new visualization model for the design of effective comparative visualization solutions for spatial data ensembles, where

- the available visualization space is exploited optimally and
- the representation of the ensemble members is abstracted such that selected spatial data characteristics are preserved optimally.

Our model helps to find an optimal compromise between the spatial abstraction of individual ensemble members and the composition of an effective comparative visualization. Due to the assistance by computational optimization, new visualization solutions are possible by simply changing one of the input parameters of the design setup. Thereby, a swift exploration of multiple, different comparative visualization options is possible.

Maybe the most important conclusion of this work is that integrating optimization into visualization clearly is an interesting and promising approach. While this is maybe not all new, it still seems worthwhile to emphasize that this way of thinking can likely lead to better visualization solutions, also in other cases.

Even though the described model is sufficiently powerful to enable a large variety of useful comparative visualization solutions, still several interesting options for future work are identified.

One way to further improve our approach would be to better steer the minimization process for finding the optimal abstraction—our current solution is based on sampling the available abstraction functions. Steering this optimization process to more efficiently minimizing the residual function would be useful, of course.

Another opportunity would be to encode (or learn), which abstraction functions preserve which spatial characteristics so that their (possibly costly) evaluation can be avoided during optimization.

In our current realization, the system of the measurements needs functions that map to the domain of the real numbers. For a number of characteristics this is not ideal. An extension of the set of measurement functions to discrete types, for example, and the error evaluation between different function types would allows us to set up even more advanced characteristics to be preserved.

Another potential limitation is the Cartesian composition of the comparative visualization. We can see the benefits of non-Cartesian composition approaches for a more advanced exploration of the data. In video visualization [5, 7] of that kind, for example, where frames are stacked onto each other, and deformed, the user is able to explore the changes in a video in one picture.

Currently, we are also working on the application of our model to a new application domain (also from biology), where large ensemble based on imaging technology are of interest.

ACKNOWLEDGMENTS

This work has been carried out within the PhysioIllustration project (#218021), which is funded by the Norwegian Research Council. We thank our application domain expert in the field of molecular biology, Mathias Ziegler, for his useful ideas and feedback.

REFERENCES

[1] O. S. Alabi, X. Wu, J. M. Harter, M. Phadke, L. Pinto, H. Petersen, S. Bass, M. Keifer, S. Zhong, C. Healey, and R. M. Taylor. Comparative visualization of ensembles using ensemble surface slicing. In *Proceedings of SPIE / Visualization and Data Analysis 2012*, pages 82940U–82940U–12, 2012.

[2] M. Alfonseca and A. Ortega. Determination of fractal dimensions from equivalent l systems. *IBM Journal of Research and Development*, 45(6):797–805, 2001.

[3] P. Angelelli and H. Hauser. Straightening tubular flow for side-by-side visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2063–2070, 2011.

[4] P. Angelelli, S. Oeltze, C. Turkay, J. Haasz, E. Hodneland, A. Lundervold, A. J. Lundervold, B. Preim, and H. Hauser. Interactive visual analysis of heterogeneous cohort study data. *IEEE Computer Graphics and Applications*, 34(5):70–82, 2014.

[5] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics*, 29(4):89:1–89:9, 2010.

[6] S. Busking, C. P. Botha, and F. H. Post. Direct visualization of deformation in volumes. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, pages 799–806, 2009.

[7] G. Daniel and M. Chen. Video visualization. In *IEEE Visualization 2003*, pages 409–416, 2003.

[8] C. Demiralp, C. Scheidegger, G. Kindlmann, D. Laidlaw, and J. Heer. Visual embedding: A model for visualization. *IEEE Computer Graphics and Applications*, 34(1):10–15, 2014.

[9] S. H. Eikeland. Multi-volume visualization and exploration. Master's thesis, University of Bergen, Bergen, Norway, 2010.

[10] J. Gain and D. Bechmann. A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics*, 27(4):107:1–107:21, 2008.

[11] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[12] J. Hocking. *Unity in Action*. Manning Publications, Shelter Island, New York, USA, 2015.

[13] W.-H. Hsu, J. Mei, C. D. Correa, and K.-L. Ma. Depicting time evolving flow with illustrative visualization techniques. In *Arts and Technology*. Springer Berlin Heidelberg, 2010.

[14] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *2012 IEEE Pacific Visualization Symposium*, pages 1–8, 2012.

[15] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009.

[16] A. Kanitsar, R. Wegenkittl, D. Fleischmann, and M. E. Gröller. Advanced curved planar reformation: Flattening of vascular structures. In *IEEE Visualization 2003*, pages 43–50, 2003.

[17] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 2013.

[18] G. Kindlmann and C. Scheidegger. An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2181–2190, 2014.

[19] P. Klemm, S. Oeltze-Jafra, K. Lawonn, K. Hegenscheid, H. Volzke, and B. Preim. Interactive visual analysis of image-centric cohort study data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1673–1682, 2014.

[20] B. Köhler, M. Meuschke, U. Preim, K. Fischbach, M. Gutberlet, and B. Preim. 2d plot visualization of aortic vortex flow in cardiac 4D PC-MRI data. In *Bildverarbeitung für die Medizin 2015*, pages 257–262, 2015.

[21] I. Kolesar, J. Parulek, I. Viola, S. Bruckner, A.-K. Stavrum, and H. Hauser. Interactively illustrating polymerization using three-level model fusion. *BMC bioinformatics*, 15(345):1–16, 2014.

[22] P. R. Krekel, E. R. Valstar, J. de Groot, F. H. Post, R. G. H. H. Nelissen, and C. P. Botha. Visual analysis of multi-joint kinematic data. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, pages 1123–1132, 2010.

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

[24] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[25] M. J. McGuffin and J.-M. Robert. Quantifying the space-efficiency of 2D graphical representations of trees. *Information Visualization*, 9(2):115–140, 2010.

[26] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[27] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 397–410, 1996.

[28] Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 301–308, 2001.

[29] D. Ritchie, S. Lin, N. D. Goodman, and P. Hanrahan. Generating design suggestions under tight constraints with gradient-based probabilistic programming. *Computer Graphics Forum*, 34(2):515–526, 2015.

[30] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2366–2375, 2013.

[31] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[32] M. Termeer, J. O. Bescós, M. Breeuwer, A. Vilanova, F. Gerritsen, and M. E. Gröller. CoViCAD: Comprehensive visualization of coronary artery disease. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1632–1639, 2007.

[33] M. Termeer, J. O. Bescós, M. Breeuwer, A. Vilanova, F. Gerritsen, M. E. Gröller, and E. Nagel. Visualization of myocardial perfusion derived from coronary anatomy. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1595–1602, 2008.

[34] J. J. van Wijk. The value of visualization. In *IEEE Visualization 2005*, pages 79–86, 2005.

[35] V. Verma and A. Pang. Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624, 2004.

[36] M. Voigt, S. Pietschmann, L. Grammel, and K. Meißner. Context-aware recommendation of visualization components. In *The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW)*, pages 101–109, 2012.

[37] T. Vrtovec, B. Likar, and F. Pernuš. Automated curved planar reformation of 3D spine images. *Physics in Medicine and Biology*, 50(19):4527–4540, 2005.

[38] J. Waser, A. Konev, B. Sadransky, Z. Horvath, H. Ribicic, R. Carnecky, P. Kluding, and B. Schindler. Many plans: Multidimensional ensembles for visual decision support in flood management. *Computer Graphics Forum*, 33(3):281–290, 2014.

[39] H. Wickham and H. Hofmann. Product plots. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2223–2230, 2011.

[40] G. Wills and L. Wilkinson. AutoVis: Automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[41] J. Woodring and H. W. Shen. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.