


Interactive Visual Similarity Analysis of Measured and Simulated Multi-field Tubular Flow Ensembles

Simon Leistikow^{1,2} , Ali Nahardani^{2,3}, Verena Hoerr^{2,3}, and Lars Linsen¹

¹Institute of Computer Science, Department of Mathematics and Computer Science, Westfälische Wilhelms-Universität Münster, Muenster, Germany

²Center for Sepsis Control and Care, Jena University Hospital, Jena, Germany

³Institute of Diagnostic and Interventional Radiology, Medical Physics Group, University Hospital Jena, Jena, Germany

Abstract

Tubular flow analysis plays an important role in many fields, such as for blood flow analysis in medicine, e.g., for the diagnosis of cardiovascular diseases and treatment planning. Phase-contrast magnetic resonance imaging (PC-MRI) allows for non-invasive in vivo-measurements of such tubular flow, but may suffer from imaging artifacts. New acquisition techniques (or sequences) that are being developed to increase image quality and reduce measurement time have to be validated against the current clinical standard. Computational Fluid Dynamics (CFD), on the other hand, allows for simulating noise-free tubular flow, but optimization of the underlying model depends on multiple parameters and can be a tedious procedure that may run into local optima. Data assimilation is the process of optimally combining the data from both PC-MRI and CFD domains. We present an interactive visual analysis approach to support domain experts in the above-mentioned fields by addressing PC-MRI and CFD ensembles as well as their combination. We develop a multi-field similarity measure including both scalar and vector fields to explore common hemodynamic parameters, and visualize the evolution of the ensemble similarities in a low-dimensional embedding. Linked views to spatial visualizations of selected time steps support an in-detail analysis of the spatio-temporal distribution of differences. To evaluate our system, we reached out to experts from the PC-MRI and CFD domains and summarize their feedback.

1. Introduction

Tubular flow refers to flow of liquids within tubular structures. A prominent example is the flow of blood within vascular structures. Analyzing blood flow is of relevance to diagnose cardiovascular diseases and to plan respective treatments. Measurement techniques such as phase-contrast magnetic resonance imaging (PC-MRI) are increasingly used to capture spatio-temporal information about blood flow. The imaging allows for a non-invasive data acquisition. The analysis of derived hemodynamic parameters can provide a biomarker to indicate abnormal blood flow [HKK*16]. However, the acquired imaging data still have low spatio-temporal resolutions and are prone to noise and artifacts such as partial volume averaging effects. The artifacts may lead to incorrect velocities, especially at vessel walls, as the observed regions remain at a sub-voxel scale. To diminish such drawbacks, new acquisition techniques (or sequences) are being developed. The goal is to improve the spatial and temporal resolution while preserving a reasonable scanning time. Each sequence has its own weakness and strength compared to others and needs to be validated against the current clinical standard, especially with respect to derived hemodynamic parameters. Such a comparison can be a tedious task, but is mandatory for methodological approval. Computational Fluid Dynamics (CFD), on the other hand, can generate noise-free simulations of

high spatio-temporal resolution. The outcome is, however, based on a larger set of parameters. In order to use the CFD simulation outcome for the analysis of a measured data set, the parameters need to be tuned to have the outcome match the measured data, e.g., to perform data assimilation. In a multi-parameter setting, this can be a complex and challenging task.

The stated problems and tasks were motivated by our collaboration partners from the respective fields. We incorporate their input to develop and present a visual computing pipeline to support domain experts in three scenarios: (1) Compare measured data acquired using different imaging settings to observe their impact. (2) Compare simulation runs generated using different parameter settings to observe their impact. (3) Compare measured data to simulated data to detect best matches in the context of data assimilation.

Such comparisons require us to analyze similarities in multi-field tubular flow ensembles. Thus, core to our visual computing pipeline is a similarity-based visual ensemble analysis. The analysis goals can be summarized as: (1) Analyzing the impact of (imaging and simulation) parameters on the tubular flow. (2) Detection of spatial regions of interest with high flow variations within the ensemble or between ensemble members. (3) Refined similarity analyses for spatial regions of interest.

To generate simulation ensembles, boundary geometry and initial conditions can either be provided or extracted from the imaging data (cf. Section 3.1), which are fed to a simulation model to allow for data-driven simulations. Then, we sample the multi-dimensional parameter space to run the simulation model with different parameter settings (cf. Section 3.2). For the comparative visual analysis of the ensemble data, we need to capture their differences or similarities. Our analysis is based on a novel multi-field similarity measure for combined scalar and vector fields (cf. Section 3.3). Our novel similarity measure is evaluated for simple synthetic data with known ground truth in Section 4. Based on the similarity measure, we developed an interactive visual analysis approach that uses linked views between similarity-space embeddings and volume visualizations to support the analysis goals listed above, see Section 3.4.

To evaluate our system, the methods are applied to and evaluated on in-vitro flow imaging data acquired using PC-MRI and flow phantoms of different geometrical complexity. For the CFD simulations we used a Lattice-Boltzmann method. Analysis results are presented in Section 5. We also reached out to experts from the PC-MRI and CFD domain and let them use our tool. We summarize their feedback in Section 6. The methods and feedback are discussed in Section 7.

Our main contributions can be summarized as follows: (1) We developed a novel, general scheme to compute multi-field similarity, which we use to combine scalar and vector field similarity. (2) We embed the scheme into a visual computing pipeline for the analysis of measured and simulated blood flow. (3) We demonstrate how our tool can support experts from the MRI and CFD domain to analyze their data by conducting two case studies.

2. Related work

We group related work in analysis methods for tubular flow data obtained via PC-MRI and CFD data, ensemble visualization methods, and multi-field similarity approaches.

Analysis of PC-MRI and CFD data. Sampling a parameter space according to fitted statistical models and *generating simulation ensembles* with the sampled parameter values is a common approach to generate ensembles. For example, in the HemeLB [MC08] framework, simulation settings were configured based on statistical data (such as pulse and blood pressure) taken from different patients. For each setting a simulation was generated. The outcome was compared using the location and intensity of wall shear stress (WSS). This approach, however, was not based on blood flow imaging data. Jiang et al. [JJVS*11], instead, acquired 4D PC-MRI data and applied a segmentation step to obtain input geometry for performing several simulation runs (using a Navier-Stokes solver). As before, the data sets were compared using WSS. High correlations between simulated and measured data could be observed. As WSS is derived from velocity fields, which may suffer from imaging artifacts, we believe it would be better to perform the comparison directly on the velocity fields and not solely on derived features, while hemodynamic parameters such as WSS or pressure can be investigated as additional features.

Ensemble visualization. Our approach is based on the concept of ensemble visualization. Ensembles were used in a wide range of scientific applications, one of which is the analysis of weather and climate data. One main motivation for generating simulation ensembles is due to complex models, which cause high data uncertainty. Thus, tools have been developed that can solve or are tailored for specific needs such as predicting hurricane trajectories [PWB*09] or generating atmospheric probability models [PWH11]. Respective approaches in ensemble visualization often try to aggregate the simulation runs to obtain statistical information such as the median and standard deviation around the median or outliers, which can then be visualized, e.g., by rendering bands or selected representatives [WHL18]. Our primary goal is to compare simulation runs, i.e., aggregating information across simulation runs is only of interest as a secondary analysis goal. Demir et al. [DDW14] create a linearization of space to generate multi-level representations of 3D ensembles. At each level, a single bar in a multi-chart represents statistical properties such as correlation between ensemble members within the respective 3D sub-domain. This approach, however, is restricted to a single scalar field per member. Additionally, making use of a Monte Carlo sampling strategy could be a more effective way to achieve an overview visualization of the whole domain than by starting at a coarse resolution, we believe.

Since we are dealing with flow data, we consider vector fields in addition to scalar fields. Jarema et al. [JDKW15] use a comparative approach to analyze non-temporal 2D vector field ensembles which aims to quickly identify the most similar groups of ensemble members and those of high variation. Hierarchical clustering is used to group pairs of members according to their similarity and glyphs in a spatial representation. We, instead, make use of 3D vector field ensembles, where glyph renderings would suffer from occlusion. Instead we analyze similarity among ensemble members, where we can build upon the comparative visualization approach proposed by Fofonov et al. [FML15]. They use similarity embeddings of individual ensemble members for comparative views. However, their approach is restricted to scalar fields. Hence, we extend their work by proposing a multi-field similarity measure for combined scalar and vector field similarity analysis. To address the spatial distribution of difference for a single time step in particular, we identify groups by performing a connected component analysis [IS09] on the per-field voxel-wise standard deviation using a user-defined threshold.

Multi-field Similarity Measures. When assuming scalar fields, many similarity measures exist based on correlation [Don03] or gradients [EHN04, NNN11, STS06]. Fofonov and Linsen [FL19] proposed a measure as an extension to isosurface similarity, which they showed to be superior for certain applications. We built upon this for handling scalar fields. They also extended their measure to multi-fields, but excluding vector fields.

For vector fields one can define the dissimilarity of vector fields by calculating the averaged and also normalized magnitude of vector differences

$$d_{diff} = \frac{\sum_{i=1}^N |\mathbf{u}_A(\mathbf{p}_i) - \mathbf{u}_B(\mathbf{p}_i)|}{2Nu_{max}}$$

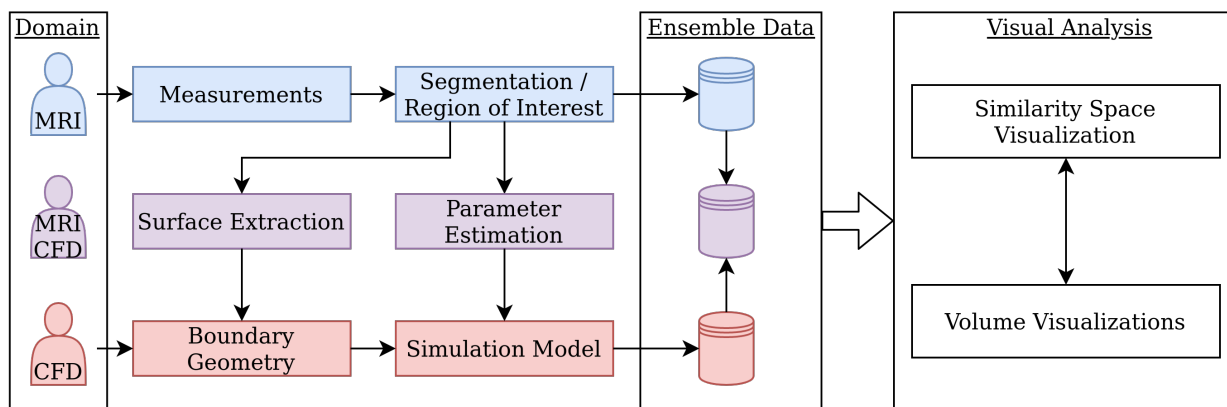


Figure 1: Visual Computing Pipeline: Geometry and flow parameters are extracted from tubular flow measurements for generating flow simulations. Ensemble of (measured and simulated) flow data is fed to the similarity-based visual analysis linking similarity-space and volume visualizations.

of the respective velocity fields \mathbf{u}_A and \mathbf{u}_B . Here, u_{max} denotes the maximum velocity magnitude of all vector fields. Jiang et al. [JJVS*11] used a weighted metric

$$d_{Jiang}(\mathbf{u}_A, \mathbf{u}_B) = \alpha e^{-2 \arcsin\left(\frac{|\mathbf{u}_A \cdot \mathbf{u}_B|}{|\mathbf{u}_A| \cdot |\mathbf{u}_B|}\right)} + \beta e^{-\frac{|\mathbf{u}_A| - |\mathbf{u}_B|}{u_{max}}}$$

with $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$ has already been used by the authors to compare simulations with 4D-PC-MRI data, where $\alpha = \beta = 0.5$ was chosen. We propose novel measures that separates magnitude and orientation.

Moreover, we propose a general scheme to generate multi-field similarity. In principle, any metric can be applied within our scheme including the metrics mentioned here. However, for our visual analysis we aim for simplicity and want to reduce the number of parameters that need to be chosen manually. We nevertheless have integrated the presented metrics into our system.

3. Visual Analysis of Ensemble Data

Our goal is to support domain experts from the MRI and CFD domain with a comprehensive system that supports all processing steps from handling measured data over setting up the simulation ensemble to the similarity-based analysis. To address respective data and tasks, we developed a visual computing pipeline, depicted in Figure 1. As for the MRI domain, our system provides interactive segmentation and cropping methods to focus the visual analysis on a region of interest. Simulation ensembles can be generated given a simulation boundary geometry and a multi-parameter simulation method. The measured data can be used to extract a boundary geometry and to estimate parameters (Section 3.1) which can be fed to the simulation model to create data-driven simulations (Section 3.2). The result is an ensemble of imaging data, simulation data, or their combination, which is then visually analyzed (Section 3.4) based on the proposed multi-field similarity measure (Section 3.3).

3.1. Measured Data Processing

In a first step, we need to *extract the boundary geometry* for the flow from the given imaging data. If an anatomical image is acquired at high spatial resolution and high signal-to-noise ratio, it is advisable to use that as input. In general, any measured scalar or vector field can be used, where in the case of a vector field, one can simply convert it to a scalar field by taking the magnitudes. As the segmentation of the flow volume serves as simulation domain, a high-quality segmentation result is desired. We apply a customizable visual processing pipeline, which depending on the image quality includes cropping, thresholding, and morphology operations combined with median and Gaussian filtering to reduce noise and undesired artifacts and to smooth the geometry. To further improve the segmentation, we also provide a probabilistic semi-automatic random-walker implementation [PRH10]. Here, the user needs to place foreground and background labels on arbitrary slices of the input volume by a simple, integrated freehand drawing tool. This can be especially useful if the contained geometry is too complex for a segmentation by the above-mentioned image operations. A subsequent binarization step leads to a binary mask, from which an isosurface is extracted using the Marching Cubes algorithm [LC87] to obtain the boundary geometry. As the simulation method requires a water-tight mesh, we automatically close holes in the geometry (if existent). To do so, we convert the isosurface triangle soup to a half-edge data structure to reconstruct connectivity information. Redundant vertices are removed using a fast neighbor-search octree implementation [BSC15]. Exploiting the half-edge data structure, we find edges without adjacent faces and loop around the boundary of each hole until we reach the starting edge. The arithmetic mean of all vertex positions on the connected path is added to the geometry and connected to the surrounding vertices to fill the hole.

In a second step, we need to *define the in- and outlets* of the flow, which are required for the definition of simulation boundary conditions. We re-use the acquired binary mask for a centerline extraction from which we automatically derive initial in- and outlets. Each endpoint of the resulting set of connected segments that is not at a bifurcation is considered a candidate for an initial in- or outlet.

We define a local disk-shaped area with radius being the respective radius of the segment. Its orientation is simply determined by the vector from the preceding point on the centerline to the considered point. If a measured flow field is provided, it is sampled within the disk to determine the normalized average local flow direction to automatically set the candidate to either in- or outlet. This initial automatic assignment can be adjusted interactively by observing the extracted geometry and detected in- and outlets in a 3D interactive visual representation as in Fig. 2. Additional disks can be arbitrarily placed along the centerline and might be used to sample the flow's cross-section at certain regions of interest which can be visualized using simple line plots depicting the evolution over time.

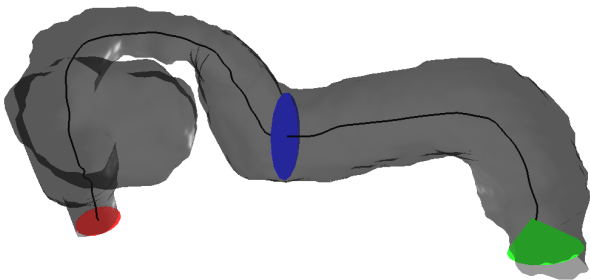


Figure 2: Interactive 3D rendering of the extracted geometry, its centerline (black) and detected inlet (green) and outlet (red). Additional disks (blue) can be placed along the centerline to analyze the local flow profile. The pump's hose has been cropped.

3.2. Simulation Ensemble Generation

Given a simulation boundary geometry as well as inlets and outlets for the flow and a multi-parameter simulation method, we can generate a simulation ensemble. The geometry hereby is either provided by a domain expert or has been acquired from the measured data (cf. Section 3.1). In principle, any method could be used that can simulate tubular flow. In this paper, we apply Lattice-Boltzmann methods [KKK*17] using a Bouzidi [BFL01] boundary condition, both implemented by the OpenLB framework [KMT*18] to generate simulation runs. The implementation discretizes the domain into a regular grid and has been shown to accurately handle even irregular geometries at appropriate resolutions, while being efficient by exploiting modern parallel hardware [HHKR12]. The relevant simulation parameters and their respective value ranges depend on the actual problem, assumptions made, and the model used. Typical parameters for tubular flow are the fluid's density ρ and viscosity ν . A common model constant for Lattice-Boltzmann methods is the Smagorinsky constant [MHK*17] that affects the behavior of turbulent flow. We start with identifying plausible value ranges with respect to the problem. For the fluid's density and viscosity, we derive the range from common literature values. For the Smagorinsky constant C_S typically positive values in the order of 10^{-1} with $C_S \geq 0.1$ are chosen [MHK*17], which we adopt for our purposes. Higher values hereby lead to more straightened turbulence. The peak flow velocity at the inlet may also be varied, but it can also be derived from measured data as described in Section 3.1.

Apart from the simulation parameters, there are also numerical parameters. These are grid constant Δx and relaxation time τ^* , which define the spatial and temporal resolution of the simulation. The characteristic velocity u_m defines the highest expected simulated velocity, which is crucial for stability and usually requires an educated guess. An initial value can also be derived from measured data, e.g., by setting it to $VENC$ which refers to the velocity encoding in MRI measurements. Additionally, further constraints need to be satisfied in order to generate stable simulations, i.e.,

$$\frac{1}{3} \frac{\rho}{\nu} u_m (\tau^* - 0.5) \Delta x < 0.4$$

needs to be fulfilled [KKK*17].

To generate the ensemble, we need to sample the simulation parameter ranges. Sampling can be performed with respect to a probability distribution. For example, if an average value is known together with an uncertainty in the form of the standard deviation or variance, we can use a Gaussian kernel to obtain a normal distribution. Since we mainly had intervals for parameter ranges, we selected a uniform sampling leading to an equal distribution of samples within the given interval for the examples presented in this paper. Parameter ranges, distributions, and number of samples per parameter dimension can be chosen by the user. Then, we generate a simulation run for each sample of the parameter space. We execute the ensemble generation on a compute cluster. One may also follow an adaptive sampling strategy, where after an initial analysis further runs are generated in regions of interest, but one has to consider that running the simulation requires some time, even when executed on the cluster. To reduce the amount of unstable simulations in advance, our system gives immediate feedback, if a parameter setting does not fulfill the above-mentioned condition.

3.3. Dissimilarity Measure

In the following we present a similarity measure for combined scalar and vector fields for comparing the time steps of the ensemble data. A time step is initially given as the velocity field \mathbf{u} at a simulated or measured point in time. Such a vector field has two characteristics, its magnitudes and its directions, each of which can be transformed into separate three-dimensional scalar fields by calculating the magnitudes $|\mathbf{u}|$ and angle differences ϕ (with respect to another velocity field). Additional hemodynamic parameters can be derived from the velocity field, both scalar and vector-valued such as wall shear stress (WSS) magnitude or vorticity, respectively [HKK*16].

For measuring *scalar field similarity*, Fofonov and Linsen [FL19] have already defined an effective distance metric, which we adopt. It is based on a generalization of the comparison of two isocontours, each extracted from a scalar field. The applicability of this method has already been well investigated and confirmed [FL18]. A Monte Carlo sampling is chosen, in which N random samples are selected within the common domain. The number of data points considered can thus be reduced significantly without losing important properties. Fofonov and Linsen reported a sample size of $N = 16,384$ to be sufficient for all simulations they encountered, which they showed by the convergence of differences in distances when linearly increasing the number of data points.

Further experiments on other volumetric data sets support the finding [FL18, LHFL19]. In general, the actual number of samples required is highly dependant on the image resolution. Since our simulation space is restricted to be inside the boundary geometry, even a number may suffice. As defining a random sampling inside a complex geometry is difficult, we generate random samples within the bounding volume of the geometry but only consider those samples that fall inside the volume that the geometry bounds.

Let \mathbf{p}_i , $i = 1, \dots, N$, be the Monte Carlo sample points and $c_i = (|\mathbf{u}(\mathbf{p}_i)| - u_{min}) / (u_{max} - u_{min})$ be the respective scalar values normalized by the value range $[u_{min}, u_{max}]$ of magnitudes of all fields. By testing each c_i against an iso-value v , isocontours can be defined, which define sets of sample points lying inside the isocontours ($c_i \geq v$). The Jaccard index is an effective measure for calculating the similarity of two sets A and B of sample points. An isosurface dissimilarity or distance metric is then given by

$$d_{AB} = 1 - \frac{|A \cap B|}{|A \cup B|}.$$

Fofonov et al. generalized the metric arguing with infinitely many isocontours to

$$d_{AB} = 1 - \frac{\sum_{i=1}^N 1 - \max(c_i^A, c_i^B)}{\sum_{i=1}^N 1 - \min(c_i^A, c_i^B)} \quad (1)$$

and showed that it has more desirable properties for defining embeddings than other metrics based on gradients or correlation.

If we now want to consider the *vector field similarity* of the velocity directions $\frac{\mathbf{u}}{|\mathbf{u}|}$ of two respective vector fields \mathbf{u}_A and \mathbf{u}_B , a scalar field distance metric can no longer be applied. Instead, a metric needs to be defined which maps the dissimilarities $\varphi_{AB}(\mathbf{p})$ of two respective direction vectors $\frac{\mathbf{u}_A}{|\mathbf{u}_A|}$ and $\frac{\mathbf{u}_B}{|\mathbf{u}_B|}$ at a point \mathbf{p} to the interval $[0, 1]$. As already stated, a straight-forward mapping to use is the angle between the directions. Then, the metric can be described as follows:

$$\varphi_{AB}(\mathbf{p}) = \begin{cases} \frac{1}{\pi} \arccos\left(\frac{\mathbf{u}_A(\mathbf{p}) \cdot \mathbf{u}_B(\mathbf{p})}{|\mathbf{u}_A(\mathbf{p})| \cdot |\mathbf{u}_B(\mathbf{p})|}\right) & \text{if } |\mathbf{u}_A(\mathbf{p})| \neq 0 \wedge |\mathbf{u}_B(\mathbf{p})| \neq 0 \\ 0 & \text{if } |\mathbf{u}_A(\mathbf{p})| = 0 \wedge |\mathbf{u}_B(\mathbf{p})| = 0 \\ 1 & \text{otherwise.} \end{cases}$$

Thus, if the velocity in both fields is non-zero, the angle between the two direction vectors, normalized to the interval $[0, 1]$, is computed. If both input velocities are zero, then the fields are maximally similar at point \mathbf{p} and the dissimilarity metric evaluates to 0 (identity-of-indiscernibles property of a metric). If exactly one of the two velocities is zero, the metric evaluates to the maximum dissimilarity 1. A zero velocity inside the simulation domain is expected to be a rare case, even more so in the measured data due to the noise. Hence, such a case would be an outstanding case that should be detected and given attention. In order to combine the results of all sample points into a single metric, we calculate the mean dissimilarity

$$d_{\angle(A,B)} = \frac{\sum_{i=1}^N \varphi_{AB}(\mathbf{p}_i)}{N}. \quad (2)$$

Having defined distance (or dissimilarity metrics) for scalar and vector fields, a *multi-field distance metric* can be defined by combining the metrics. Assuming that we want to combine distances

obtained by n applications of one of the distance metrics on a scalar or vector field, we need to define a function f to combine them. The fact that both metrics are defined on the interval $[0, 1]$ can be exploited for their combination. Function f shall have the properties

$$f(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } \exists i : x_i = 1 \\ 0 & \text{if } \forall i : x_i = 0 \\ 0 \leq \gamma \leq 1 & \text{otherwise} \end{cases}$$

such that a consistent interpretation of individual and combined distance metrics is possible. Moreover, $f(x_1, \dots, x_n)$ (and thus γ) shall be a monotonically increasing function for all inputs x_i (within the interval $[0, 1]$). Thus, if all distance metrics x_j , $j \neq i$, are constant and x_i increases, then $f(x_1, \dots, x_n)$ shall also increase (or remain constant). One possible choice is to set

$$f_{max}(x_1, \dots, x_n) = \max(x_1, \dots, x_n). \quad (3)$$

However, only the entry representing the greatest dissimilarity of all distance metrics is used. This is useful, if one is looking for ensemble members that are similar in all fields. The disadvantage, however, is that a lot of information gets lost. In particular, it is not clear whether only one field is responsible for the dissimilarity, or several. Therefore, to equally include all field dissimilarities, we propose to use the function

$$f_{prod}(x_1, \dots, x_n) = 1 - \prod_{i=1}^n (1 - x_i). \quad (4)$$

The product additionally penalizes high dissimilarity across all fields, which was found to be a desirable property for the embedding, especially when compared to computing the mean dissimilarities. Moreover, it can be easily checked that all desired properties are fulfilled. Which of the two combination rules to choose depends on the analysis task.

The metrics proposed and discussed above will be further evaluated in Section 4.

3.4. Visual Analysis

Similarity-space visualization. Our visual analysis component consists of multiple linked views, but the main view is the *similarity-space visualization*, which is supposed to exhibit the similarity of the ensemble members including their temporal evolution (if available). Thus, we want to develop a lower-dimensional embedding for the whole ensemble of measured and simulated data, where dissimilarities between time steps are represented by Euclidean distances in the embedding. Each time step of each ensemble member is represented by one point in the embedding. If two points are close in the embedding, the fields of the corresponding ensemble member time steps should be similar with respect to the chosen similarity measure. To analyze the temporal evolution, we connect the embedded points belonging to one time series in chronological order to form curves that are parametrized over time.

To generate the embedding, we first need to define the similarity space, i.e., we have to decide which of the dissimilarity metrics of the previous section to choose. Having defined the metric, we can compute pairwise similarities among all time steps of all

ensemble members (possibly including both measured and simulated data). The pairwise dissimilarities can be stored in a symmetric distance matrix. Given the distance matrix, we follow the approach of multi-run plots proposed by Fofonov et al. [FML15], where the distance metric is fed to a classical multidimensional scaling (MDS) approach [Wic03]. Since our objective is to have the Euclidean point distances in the embedding match the dissimilarities stored in the distance matrix, the objective function is that of an MDS approach such that we should prefer an embedding computed by MDS over one of the many other dimensionality reduction approaches (cf. [SNAA19, NA19]).

Using classical MDS, the axes of the projection space are the principal directions with corresponding eigenvalues. Only those with large eigenvalues contribute significantly to the projection. Hence, the number of principal directions used is determined by the number of significant eigenvalues. In practice, the dimensionality of a visual space is restricted to, at most, 3 dimensions. Thus, we use the first three principal directions to create 1D, 2D, or 3D similarity plots using the time axis as second axis in the 1D case. To decide, which dimensionality to pick for the embedding, we perform an eigenanalysis, which is supported by plotting a bar chart with largest eigenvalues. The intrinsic dimensionality can be deduced, which can then be used for the embedding. If the intrinsic dimensionality is larger than three, we could interactively swap axes to look into more than 3 principal directions (cf. [FML15]), but this was not necessary for the examples presented in this paper.

To each ensemble member a unique color is assigned according to a user-defined color map, which is used to color code the resulting time lines. To add back time information when using 2D or 3D embeddings, the members' respective color changes over time, linearly transitioning towards white. Ensemble members with no temporal component represent static data. Such members are depicted as horizontal lines in the 1D case and as spheres in the 2D and 3D case, respectively. Examples of 1D, 2D, and 3D plots are presented in Figures 3(a), 3(b), and 3(c), respectively.

The similarity-space embedding provides a natural starting point for our *analytical workflow*. All ensemble members are loaded to the visualization for further investigations. The eigenvalue bar-charts are used to select the appropriate dimensionality of the embeddings, but the user may also switch between 1D, 2D, or 3D embeddings. Similarly, the user may also select a respective similarity measure. Further, a subset of ensemble members can be selected, for which the embedding is recalculated. This is especially useful if the ensemble contains outliers that make other ensemble members clutter together. The outliers may be removed to provide more degrees of freedom for embedding the remaining members. Similarly, one can drill down on clusters and subclusters.

Volume visualizations. Having detected time steps of ensemble members that deserve an in-depth analysis, linked views to *volume visualizations* are provided. The volume visualizations of a selected time step shall allow for (i) the multi-field visualization of a single ensemble member, (ii) the comparative visualization of two ensemble members, (iii) the statistical visualization of all ensemble members. Visualizing a single ensemble member supports the in-depth analysis of a member that has been identified as being interesting during the ensemble analysis. Visual comparison of two ensemble

members allows for the detection of spatial regions where two identified ensemble members differ, e.g., in which regions a simulation run fails to reproduce a measured field. The simultaneous volume visualization of all ensemble members shall support the identification of spatial regions with different ensemble behavior, e.g., detecting spatial regions where all ensemble members tend to concur or tend to disagree. Moreover, we use regions with high disagreement to define new seeds for the similarity-space embedding. By comparing the embeddings for selected regions only, the effect of sequences or simulation parameters can be analyzed.

For volume visualizations of a single ensemble member, we use direct volume rendering via (multi-modal) GPU-based raycasting and linked slice-based viewers. For the slice-based visualization, the user may use a handle metaphor to arbitrarily position a plane in the volume, from which the slice projection is derived. Interacting with such a slice view reveals the local velocity values. Additionally, the cursor's position is projected back into 3D space for a better spatial understanding of the linked views, see Figure 10(a). For velocity fields of selected time steps, streamlines (or pathlines for members with multiple time steps) can be calculated using a classical Runge-Kutta integration scheme and visualized using a velocity-magnitude or direction color-coding, see Figure 11.

For a comparative volume visualizations of any scalar or vector field of two selected ensemble members, we create a (vector) difference volume from the respective fields and provide slice-based visualization of the difference volume as well as a direct volume rendering, see Figure 10(a). Depending on the transfer function, regions of high differences can be highlighted, e.g., when using a maximum intensity projection. Alternatively, juxtaposed linked volume visualizations enable a fast comparison of two velocity fields, especially with regard to flow features such as turbulences or vortices, cf. the streamline visualizations in Figure 11.

To support volume visualization for the entire ensemble, we first calculate the mean volume of all N ensemble members for a point in time and for any selected field. For vector fields, the mean of each, x-, y- and z-component is calculated, so that each voxel is effectively represented by a mean vector $\bar{\mathbf{x}}$. We then use this average to calculate a *similarity volume* as shown in Figure 5, where we perform a voxel-wise calculation

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N |\mathbf{x}_i - \bar{\mathbf{x}}|}, \quad (5)$$

similar to computing the standard deviation. We then perform a user-defined thresholding on the scalar field of σ -values to create a binary volume, on which we run a connected component analysis [IS09]. The resulting regions are filtered and sorted in descending order according to their enclosed volume (simply estimated by the number of enclosed voxels). The boundary geometry (cf. Section 3.1) is then rendered with low opacity and overlaid with a ray-casting of the binary volumes representing the detected regions, where each region gets assigned a unique color. The resulting visualization can be seen in Figure 8, respectively. This allows for a fast assessment of the spatial distribution of difference for any field of the ensemble at a selected point in time. The methods were implemented using Voreen [MSRMH09], an open-source framework for volume visualization and will be published in an upcoming release.

4. Evaluation of Similarity Measure

To evaluate our similarity measure, we generate an ensemble of three very simple synthetic vector fields, which is summarized in Table 1. In particular, the vector fields are homogeneous and can, therefore, be described by a single vector for each time step. Since our evaluation is performed qualitatively, the vectors have been chosen to quite drastically change either magnitude, direction, or both over time. Thus, each vector depends on a parameter $\theta \in (0, 1]$ which was equidistantly sampled 40 times to simulate the change over time by an increasing parameter θ .

Member	Quantity	Vector	Color
1	Magnitude	(θ, θ)	orange
2	Direction	$(\cos(2\pi\theta), \sin(2\pi\theta))$	pink
3	Combined	$(\theta \cdot \cos(2\pi\theta), \theta \cdot \sin(2\pi\theta))$	blue

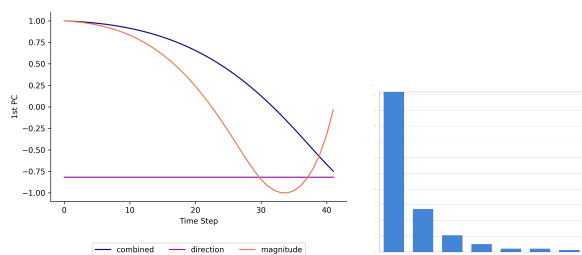
Table 1: Ensemble of the synthetic flow fields used as ground truth for our similarity measure evaluation. Each member is defined by a unique vector depending on θ which increases linearly over time on interval $(0, 1]$ using 40 samples. The last column provides the color used in the similarity plots for the runs.

First, the metrics in Equations 1 and 2 are evaluated for separate consideration of magnitude and direction. Figure 3(a) shows a 1D embedding with time as second axis using only velocity magnitude for dissimilarity (Equation 1). As expected, Member 2, which does not change its magnitude shows up as a horizontal line. If only the magnitude changes (Member 1), the corresponding curve also exhibits a drastic change. If the changes in magnitude and direction are combined (Member 3), the resulting curve behaves similarly, since the direction has no influence. However, the curves are not identical, since the magnitudes differ by a factor of $\sqrt{2}$. Member 3 approaches Member 2, as their magnitudes approach each other towards the end. The actual position at the last time step, however, is not the same as, since the second principal directions also has a non-negligible influence, which can be seen by considering the corresponding eigenvalue barchart. Member 1, on the other hand, moves away again due to its increased magnitude.

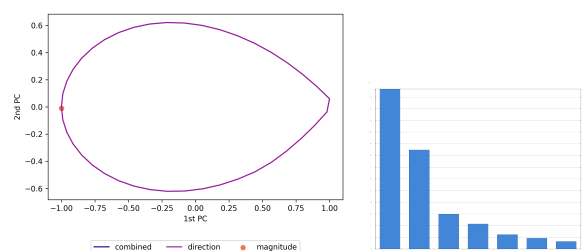
Figure 3(b) shows a 2D embedding using only velocity direction for dissimilarity (Equation 2). As expected, the circular curves of Members 2 and 3 coincide. Member 1 does not change direction and, thus, is represented by the dot.

When using our proposed combination of magnitude and direction information using the metrics in Equations 3 or 4, we observe that both magnitude and direction are captured appropriately. Figure 3(c) shows the results when using Equation 4. Here, 3D embeddings are used. We observe a circular curve on a plane for Member 2 and an almost linear path for Member 1. The curve of Member 3 clearly represents the desired combination of both aforementioned, where the circular curve of Member 2 is translated over time in the direction of the curve of Member 1 to form a spiral.

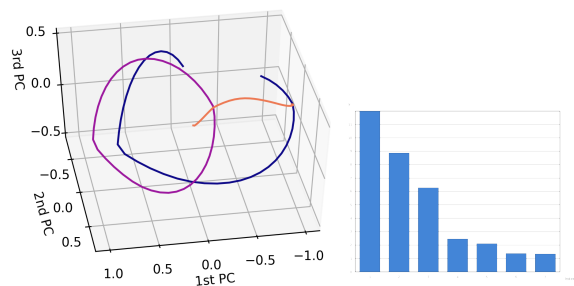
We compare our result against that of Jiang et al. [JJVS*11] as discussed in Section 2, which is shown in Figure 3(d). The circle described by Member 2 is severely distorted, while Member 1 is almost static. Consequently, Member 3 also shows up as a distorted



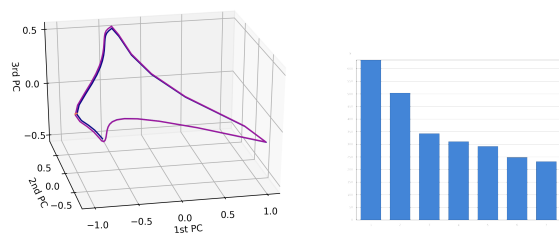
(a) 1D similarity plot for synthetic data ensemble using metric from Equation 1 only considering vector magnitude. The eigenvalue barchart indicates that using the first principal directions (viewed over time) preserves most information.



(b) 2D similarity plot for synthetic data ensemble using metric in Equation 2 only considering vector direction. The eigenvalue barchart indicates that using the first two principal directions is sufficient.



(c) 3D similarity plot for synthetic data ensemble using combined metric in Equation 4. The eigenvalue barchart indicates that using the first three principal directions is sufficient.



(d) 3D similarity plot for synthetic data ensemble using the metric by Jiang et al. The eigenvalue barchart indicates that using the first three principal directions may not be enough.

Figure 3: Similarity plots (left) and respective eigenvalue bar charts (right) using different similarity measures for the synthetic vector field ensemble.

circle that almost coincides with Member 2. Hence, we can conclude that our measure follows better the expectations.

Finally, we interpret the x , y , and z components of the vector fields as three scalar fields to compare our result against that of Fofonov et al. [FL19] using their multi-field approach for scalar fields. The result, however, does not properly reassemble the circular shape as our measure did.

5. Ensemble Analysis Results

In this section, as a proof of concept, we apply our methods to in-vitro imaging data sets acquired by a 9.4 T Bruker BioSpec small animal scanner for flow boundaries of different complexities.

5.1. Pipe Flow Phantom

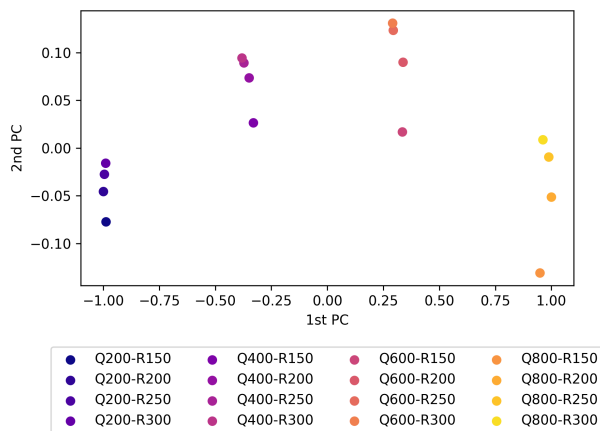


Figure 4: 2D similarity space embedding for the pipe flow phantom ensemble using our similarity measure by combining velocity magnitude and angle information by f_{prod} . Q -annotated numbers encode the pump speed in ml/min and R -annotated numbers encode the grid constant in μm .

The simplest tubular flow imaginable is laminar flow in a pipe. Hence, this example is the state-of-the-art benchmark in CFD simulations. We transfer this example to the MRI domain and try to draw conclusions using our visual analysis approach. More precisely, we use data generated by the 4D center-out stack-of-stars PC velocity mapping sequence ($VENC = 150\text{ cm s}^{-1}$) applied on a phantom made from acrylic glass, which contains four smaller tubes. A water pump was connected to two of the tubes, while the others were connected such that they define a closed circuit. For the experiments, 4 different pump speeds of 200, 400, 600 and 800ml/min and 4 different isotropic resolutions of 200, 150, 120 and 100 pixels with a respective grid constant of 150, 200, 250 and 300 μm^3 have been chosen. Each pump speed was measured in all resolutions by otherwise identical conditions to generate an ensemble with a total of 16 members for each of which an “anatomical” image and the velocity vector fields is provided. In principle, the four smaller tubes are mostly identical, however we choose to use all of them for the analysis to effectively increase our sample size.

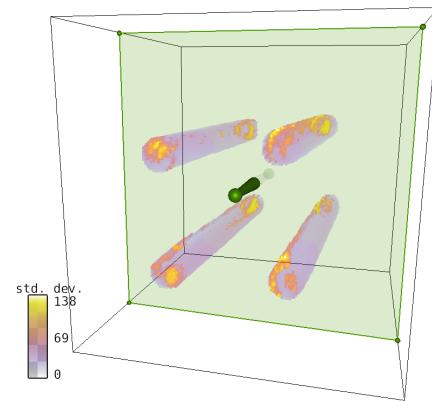


Figure 5: Similarity volume rendering for pipe flow phantom ensemble using maximum intensity projection. The green handle metaphor can be used to position an arbitrary plane from which a slice-based representation can be derived.

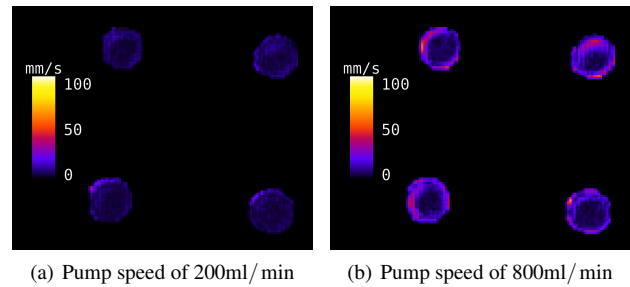


Figure 6: Cross-section through magnitude velocity vector difference volume calculated between the lowest and highest resolution for different pump speeds for pipe flow phantom. The plane was chosen as shown in Figure 5.

First, we use the magnitude images of the highest resolution to perform a segmentation using the semi-automatic random-walker implementation. The segmentation is then used, according to our proposed analysis workflow, to place seeds for the similarity embedding. Since fluctuations in pump/water speed have been averaged out over time, the measurements represent mostly steady flow, i.e., we have one time step for each ensemble member.

Using our similarity measure, we combine the dissimilarities of the velocity magnitude (Equation 1) and angle difference (Equation 2) using f_{prod} (Equation 4) to create a 2D embedding for the entire ensemble, see Figure 4. Function f_{prod} was chosen, as we want to give equal impact to both magnitude and direction. Using the first two principle components as axes, we can immediately identify 4 groups with 4 ensemble members each. The groups are formed by members obtained using the same pump speed. The groups also exhibit an order according to increasing pump speed. Within each group we observe an order based on the grid constant.

The similarity volume reveals that the highest difference across all ensemble members is found at the ends of the tubes and near

their boundary. In the PC-MRI domain, acquiring accurate flow data at the vessel walls is a common problem and thus active field of research.

Another observation from the similarity space embedding in Figure 4 is that the spread within each of the 4 groups increases with higher pump speed, i.e., their velocities presumably differ more for higher pump speeds. To assure that this is not a projection artifact of the embedding, we interactively selected pairs of highest dissimilarity among the groups with lowest and highest spread, respectively, and visualize the magnitude of the respective vector field difference volumes. We observe in Figure 6 that the differences in velocities indeed increases with higher pump speed.

5.2. Aneurysm Flow Phantom

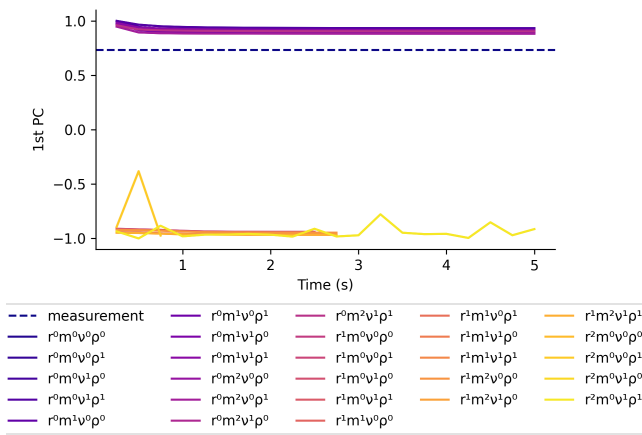


Figure 7: 1D similarity embedding for brain aneurysm ensemble. The parameters of each run are encoded by a combination of resolution r , inlet flow velocity magnitude m , viscosity v , and density ρ . The indices denote the respective value, 0 being the lowest. The bottom group represents simulation runs that failed indicated by early termination or fluctuations. The dashed line represents the (not time-varying) measured data.

To evaluate our methods on a more complex and more realistic geometry and to evaluate the effect of multiple simulation parameters, an in-vitro brain aneurysm model was measured by the use of 4D FLASH PC velocity mapping MRI sequence. The in-vitro flow phantom was 3D printed and remodeled using liquid latex to have a similar elasticity close to the human vessel wall. In our study, the experiment has been performed with a flow rate of 400 ml/min, $VENC = 300 \text{ cm s}^{-1}$, and an isotropic resolution equal to $391 \mu\text{m}^3$ resulting in an image resolution of 128^3 pixels.

First, we process the measured data as described in Section 3.1. In particular, we apply a thresholding operation and a median filter on the velocity magnitude field to extract the vessel lumen. A 3D Gaussian filter is used to obtain a smoother surface during iso-surface extraction, the result of which can be seen in Figure 2. We generate a simulation ensemble by varying parameters viscosity,

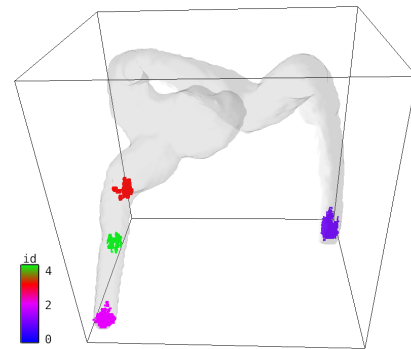


Figure 8: The 4 largest regions calculated by a connected component analysis on the similarity volume (binarization threshold = 20) for the last time step of converged simulations and measured data, generated using the brain aneurysm phantom.

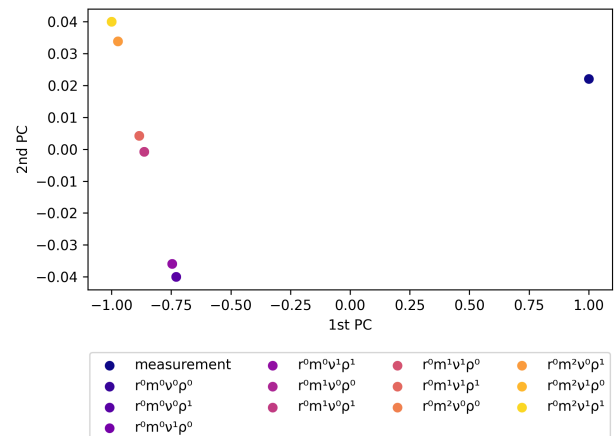


Figure 9: 2D similarity embedding, where seeds have been restricted to the green and red region in Figure 8. Plotted points overlap in the projection due to density having negligible impact on the simulation output.

density, and the inflow velocity. In our simulations, we assume the moving fluid to be blood rather than water, as blood has a wider range of viscosity ($3 \times 10^{-3} \text{ m}^2 \text{ s}^{-1} - 4 \times 10^{-3} \text{ m}^2 \text{ s}^{-1}$) and density values ($1043 \text{ kg m}^{-3} - 1057 \text{ kg m}^{-3}$). We equidistantly sample the parameter space using the respective minimum and maximum values from the intervals. The inflow velocity was chosen to be at 320 cm s^{-1} , 400 cm s^{-1} , and 480 cm s^{-1} , respectively. The Smagorinsky constant was set to 0.1, which is common for a simulation domain of this kind [MHK*17].

The simulation output is a velocity vector field similar to what is provided by the measurement. We used three different simulation resolutions, however, downsampled them to 128 voxels in each spatial dimension, which was found to provide a suitable trade-off

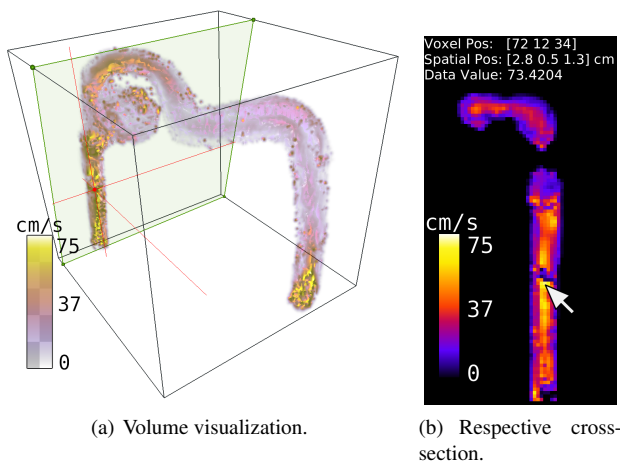


Figure 10: Velocity magnitude difference visualization of measured data and the simulation run most similar to it, after the most dissimilar regions have been removed according to Figure 8. A specific voxel value is shown by hovering the cursor over a pixel in the cross-section. The position is mapped back into 3D space in the volume visualization as indicated by the red dot and connected orthogonal lines.

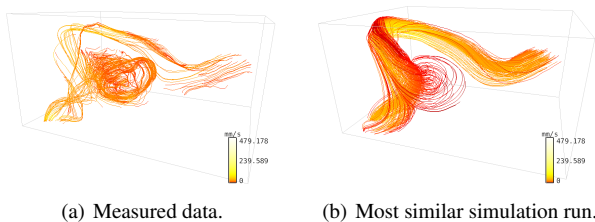


Figure 11: Juxtaposed comparative streamline visualization of measured data set and the simulation run most similar to it.

between resolving the desired flow features and allowing for an interactive analysis. Our analysis goal was to find a simulation that best matches the measurement. Consequently, we added the measurement to the generated ensemble. Using our multi-field similarity measure f_{max} (Equation 3), we combine the dissimilarities of the velocity magnitude (Equation 1) and angle difference (Equation 2) to create the 1D similarity-space visualization shown in Figure 7. We chose f_{max} because we want to find the simulation run that is most similar to the measured data in all fields. While the simulated data are spatio-temporal, the measured data have no time-varying component and, therefore, are represented by a horizontal (dashed) line. In the 1D embedding, we observe two cluttered groups, the lower of which contains unstable simulations, indicated by early termination and severe fluctuations. We therefore select the upper group only and generate a new 1D embedding which can be seen in the accompanying video. In the new embedding the simulation runs also all show a similar evolution over time. However, they differ by some offset and we were able to identify three groups consisting

of three runs each. These groups were found to have different respective inflow velocity magnitude, i.e., this parameter influences the outcome most. Each of the curves actually coincides with two other curves which correspond to different density values. Thus, we can conclude that density has a negligible impact in comparison to other parameters.

Next, we visualize the regions of high dissimilarity using the similarity volume of the selection of stable runs and the respective connected components of high disagreement shown in Figure 8. We observe four regions, which we analyze further by creating new embeddings for the last time step of each converged simulation and the measured data, where seeds have been restricted to the identified regions, see Figure 9. The red and green regions are caused by so-called phase wraps in the measurement. Phase wraps occur when measured flow velocity v is higher than $VENC$ such that it will be mapped to an incorrect velocity. For the other two regions near the inlet and outlet we observed a high variation between the simulations. It therefore seems that boundary conditions are quite sensitive to parameter changes. In order to provide a spatial understanding of the actual similarity of the velocity vector fields, we provide volume difference visualizations (cf. Figure 10) as well as juxtaposed streamline renderings (cf. Figure 11) for both the measured data set and the time step of the simulation run which is most similar according to our measure f_{max} . It was selected by considering the lowest Euclidean distance between time steps of measurement and simulations in the embedding. Additionally, we remove the already detected regions of high local dissimilarity according to Figure 8 from the difference volume to better explore the distribution of difference in the remaining space. It reveals high differences in the region where the vessel experiences high curvature, as well as right after the flow passed by the aneurysm.

6. Domain Expert Evaluation

Our tool supports experts from both the MRI and CFD domain to analyze their data. It must therefore, as a whole, be intuitive and useful to them. We reached out to three domain experts and made them use all components of the analysis workflow embedded in our tool. In the following, we summarize their feedback. The interviews were performed in one session for each domain, where two MRI experts (P1, P2) participated in the first session (S1) and one CFD expert (P3) in the second session (S2).

For S1, an ensemble of measured data similar to the one presented in Section 5.2 was provided by the experts in advance, which we used as input for our tool. Since generating simulation ensembles is not of primary interest for the MRI domain, the simulation ensemble generation was only briefly discussed. In contrast, in S2 more focus was given to this particular feature and respectively less on the measured data preparation step. During the sessions, no simulation ensembles were run, as this would have taken too much time. Nevertheless, in both sessions the experts defined parameter ranges to set up a simulation. P2 even drew one sample from the parameter space and ran this locally on his computer. For S2, we prepared the data set in Section 5.2, which already includes representative simulation data.

For the visual analysis, the experts were given an overview about

all linked views and their respective interaction. Especially the similarity embedding was explained in detail to allow for useful interpretations. Since Euclidean distances represent dissimilarities, all experts found the embedding comprehensive and intuitive, regardless of the number and order of the principal directions used. P3 reported that, in his domain, plots similar to our 1D embedding are used to analyze respective quantities of simulations like the drag coefficient. The similarity volume and the detected regions of highest dissimilarities were also rated quite intuitive by all experts.

All experts were asked about how they think the tool could be used in their daily work and what a typical use-case could look like. P1 and P2 conduct research in MRI pulse-sequence design and therefore typically perform a lot of experiments. Since the resulting imaging data needs to be compared to results from established standards, they found the tool to be a good starting point for their analysis. They especially pointed out the usefulness of the similarity embedding. However, they found that the interpretation of dissimilarity can be improved. Since Euclidean distances in the embedding only have a relative interpretation, it is currently not possible to tell without an in-depth analysis what the corresponding absolute dissimilarity is. They therefore suggested to add some statistical power to the similarity space visualization, e.g., by integrating statistical properties.

P3 instead typically deals with Lattice-Boltzmann methods in his research and claimed that it would be very beneficial for him and other members of his domain to use our tool. Especially the simulation ensemble generation caught his attention and he liked how comfortable the parameter space can be sampled which, he proposed, could be very beneficial for beginners in the simulation domain. When asked for further improvements, he suggested to integrate more tools to analyze the parameter space, e.g., by clustering the points in the similarity embedding according to their parameters. Part of his research includes adaptive grid refinement of the simulation lattice. He considers the visualization of the similarity volume and regions of high dissimilarities useful for this task. In particular, a simulation ensemble with different refinement settings could be generated and analyzed for dissimilar regions where, conclusively, the refinement would need to be adjusted.

7. Discussion

As a proof of concept, our approach was applied on two domains of different complexity. We could show that our multi-field similarity measure can be used effectively to generate low-dimensional similarity visualizations that provide a good overview for ensembles created from measured and simulation data. We created a benchmark scenario for PC-MRI sequences where laminar flow inside a flow phantom was generated by a water pump. Pump speed and resolution of the measurements were changed and we could show the effect on the measured data. Another measurement of flow inside a brain aneurysm phantom was used to extract a more complex flow geometry, for which a simulation ensemble was generated. We could determine the effect of individual parameters on the simulation outcome and compared the result to the measured data.

In the proof of concept, two fields were combined. In principle, of course, our measure is applicable to more than two fields.

In future work, we want to add further hemodynamic parameters to the analysis. The embedding is limited to up to three principal directions, however, for all the data we used to, three dimensions properly reassembled most of the data as indicated by the respective eigenvalue bar charts. Generally, our approach is independent of any measurement technique and simulation method. However, depending on the methods, the number of considered parameters might be considerably large, resulting in ensembles with a lot of members. Consequently, a large amount of simulations would need to be run and visualized in our similarity based visualization, which may cause visual clutter. To compensate for that, the similarity of intermediate time steps towards the measured data could be calculated at the time the simulations runs. Those not getting, at least, as similar as defined by a user defined threshold or even experience increased dissimilarity as time progresses could be interrupted early and removed from the ensemble. For defining such a threshold, similarity between multiple measurements or time steps of a time-varying measured data set could be used. Support for the latter, e.g., for consideration of the cardiac cycle we think is a necessary step and straight forward to incorporate in the current pipeline. Further spatial visualizations could easily be integrated such as the arrow glyph-based visualization for a pair of velocity vector fields as proposed by de Hoon et al. [dHvPJV14].

We asked domain experts from the CFD and MRI domain to provide feedback on the intuitiveness and usefulness of our methods and the application they are embedded in. The overall feedback was very positive, however, the similarity plot as an overview visualization would clearly benefit from representing absolute similarities, not only relative.

8. Conclusions and Future Work

We presented a tool for the generation and analysis of tubular multi-field ensembles and especially addressed use cases for experts from the CFD and MRI domain. The visual computing workflow includes a data processing step for measured data which can be used to initialize in- and outlets and estimate parameters for a multi-parameter simulation model. Simulation ensembles were generated using a Lattice-Boltzmann method by sampling the parameter space of relevant parameters and can be combined with measured data. We applied our similarity-based visual analysis approach to ensembles from both domains and could show that relevant properties can be extracted easily by using multiple linked views showing the spatio-temporal distribution of difference. This was confirmed by feedback of experts from the respective domains.

In future work, we want to integrate time-varying in-vitro and in-vivo measurements to achieve an even better understanding of the similarity space when compared to simulated data. As in-vivo measurements are involved, we want to not only consider static, but also deformable geometry in our workflow. Since the simulation outcome might be sensitive with regard to the simulation domain, we also want to integrate an uncertainty based sampling strategy for the geometry similarly to what was proposed by Scheid-Rehder et al. [SRLM19]. Finally, we want to address the feedback of the domain experts and advance the similarity space embedding visualization such that it can represent absolute similarity.

Acknowledgments

This work was supported by the Center of Sepsis Control & Care (CSCC); funded by the German Ministry of Education and Research (BMBF).

References

- [BFL01] BOUZIDI M., FIRDAOUSS M., LALLEMAND P.: Momentum transfer of a boltzmann-lattice fluid with boundaries. *Physics of fluids* 13, 11 (2001), 3452–3459. 4
- [BSC15] BEHLEY J., STEINHAGE V., CREMERS A. B.: Efficient Radius Neighbor Search in Three-dimensional Point Clouds. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)* (2015). 3
- [DDW14] DEMIR I., DICK C., WESTERMANN R.: Multi-charts for comparative 3d ensemble visualization. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2694–2703. 2
- [dHvPJV14] DE HOON N., VAN PELT R., JALBA A., VILANOVA A.: 4D MRI flow coupled to physics-based fluid simulation for blood-flow visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 121–130. 11
- [Don03] DONCKER P. D.: Spatial correlation functions for fields in three-dimensional rayleigh channels. *Journal of Electromagnetic Waves and Applications* 17, 6 (2003), 877–878. 2
- [EHNP04] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Local and global comparison of continuous functions. In *Proceedings of the Conference on Visualization '04* (2004), IEEE Computer Society, pp. 275–280. 2
- [FL18] FOFONOV A., LINSEN L.: A top-down interactive visual analysis approach for physical simulation ensembles at different aggregation levels. *Information* 9, 7 (2018), 163. 4, 5
- [FL19] FOFONOV A., LINSEN L.: Projected field similarity for comparative visualization of multi-run multi-field time-varying spatial data. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 286–299. 2, 4, 8
- [FML15] FOFONOV A., MOLCHANOV V., LINSEN L.: Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE transactions on visualization and computer graphics* 22, 8 (2015), 2037–2050. 2, 6
- [HHKR12] HENN T., HEUVELINE V., KRAUSE M. J., RITTERBUSCH S.: Aortic coarctation simulation based on the lattice boltzmann method: benchmark results. In *International Workshop on Statistical Atlases and Computational Models of the Heart* (2012), Springer, pp. 34–43. 4
- [HKK*16] HA H., KIM G. B., KWEON J., LEE S. J., KIM Y.-H., LEE D. H., YANG D. H., KIM N.: Hemodynamic measurement using four-dimensional phase-contrast mri: quantification of hemodynamic parameters and clinical applications. *Korean journal of radiology* 17, 4 (2016), 445–462. 1, 4
- [IS09] ISENBURG M., SHEWCHUK J.: Streaming connected component computation for trillion voxel images. In *Workshop on Massive Data Algorithmics* (2009). 2, 6
- [JDKW15] JAREMA M., DEMIR I., KEHRER J., WESTERMANN R.: Comparative visual analysis of vector field ensembles. In *2015 IEEE Conference on Visual Analytics: Science and Technology (VAST)* (2015), IEEE, pp. 81–88. 2
- [JJVS*11] JIANG J., JOHNSON K., VALEN-SENDSTAD K., MARDAL K.-A., WIEBEN O., STROTHER C.: Flow characteristics in a canine aneurysm model: A comparison of 4d accelerated phase-contrast mr measurements and computational fluid dynamics simulations. *Medical physics* 38, 11 (2011), 6300–6312. 2, 3, 7
- [KKK*17] KRÜGER T., KUSUMAATMAJA H., KUZMIN A., SHARDT O., SILVA G., VIGGEN E. M.: The lattice boltzmann method. *Springer International Publishing* 10 (2017), 978–3. 4
- [KMT*18] KRAUSE M. J., MINK A., TRUNK R., KLEMENS F., MAIER M.-L., MOHRHARD M., CLARO BARETTO A., HAUSSMANN M., GAEDTKE M., ROSS-JONES J.: Openlb release 1.2: Open source lattice boltzmann code. <https://www.openlb.net/download>, 2018. 4
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics* (1987), vol. 21, ACM, pp. 163–169. 3
- [LHFL19] LEISTIKOW S., HUESMANN K., FOFONOV A., LINSEN L.: Aggregated ensemble views for deep water asteroid impacts simulations. *IEEE computer graphics and applications* (2019). 5
- [MC08] MAZZEO M. D., COVENEY P. V.: Hemelb: A high performance parallel lattice-boltzmann code for large scale fluid flow in complex geometries. *Computer Physics Communications* 178, 12 (2008), 894–914. 2
- [MHK*17] MIRZAEI H., HENN T., KRAUSE M. J., GOUBERGRITS L., SCHUMANN C., NEUGEBAUER M., KUEHNE T., PREUSSER T., HENNEMUTH A.: MRI-based computational hemodynamics in patients with aortic coarctation using the lattice boltzmann method: Clinical validation study. *Journal of Magnetic Resonance Imaging* 45, 1 (2017), 139–146. 4, 9
- [MSRMH09] MEYER-SPRADOW J., ROPINSKI T., MENSMMANN J., HINRICHS K.: Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications* 29, 6 (2009), 6–13. 6
- [NA19] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (Aug 2019), 2650–2673. 6
- [NNN11] NAGARAJ S., NATARAJAN V., NANJUNDIAH R. S.: A gradient-based comparison measure for visual analysis of multifield data. *Computer Graphics Forum* 30, 3 (2011), 1101–1110. 2
- [PRH10] PRASSNI J.-S., ROPINSKI T., HINRICHS K.: Uncertainty-aware guided volume segmentation. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1358–1365. 3
- [PWB*09] POTTER K., WILSON A., BREMER P.-T., WILLIAMS D., DOUTRIAUX C., PASCUCCI V., JOHNSON C. R.: Ensemble-vis: A framework for the statistical visualization of ensemble data. In *2009 IEEE International Conference on Data Mining Workshops* (2009), IEEE, pp. 233–240. 2
- [PWH11] PÖTHKOW K., WEBER B., HEGE H.-C.: Probabilistic marching cubes. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 931–940. 2
- [SNA19] SAEED N., NAM H., AL-NAFFOURI T. Y., ALOUINI M.: A state-of-the-art survey on multidimensional scaling-based localization techniques. *IEEE Communications Surveys Tutorials* 21, 4 (2019), 3565–3583. 6
- [SRLM19] SCHEID-REHDER A., LAWONN K., MEUSCHKE M.: Robustness evaluation of CFD simulations to mesh deformation. 11
- [STS06] SAUBER N., THEISEL H., SEIDEL H.-P.: Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 917–924. 2
- [WHL18] WANG J., HAZARIKA S., LI C., SHEN H.-W.: Visualization and visual analysis of ensemble data: A survey. *IEEE transactions on visualization and computer graphics* (2018). 2
- [Wic03] WICKELMAIER F.: An introduction to MDS. *Sound Quality Research Unit, Aalborg University, Denmark* 46, 5 (2003), 1–26. 6