YMCA – Your Mesh Comparison Application

Johanna Schmidt, Reinhold Preiner, Thomas Auzinger, Michael Wimmer,
M. Eduard Gröller, *Member, IEEE CS*, and Stefan Bruckner *Member, IEEE CS*
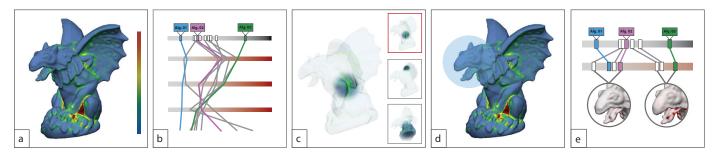
Fig. 1: Visual analysis for mesh comparison. We propose YMCA, a system that combines explicit encoding, juxtaposition and quantitative measures to allow the user to compare multiple meshes. YMCA conveys an overview of the available data (a, b), points to interesting features in the data (c) and allows for the inspection of local areas of interest (d,e).

Abstract—Polygonal meshes can be created in several different ways. In this paper we focus on the reconstruction of meshes from point clouds, which are sets of points in 3D. Several algorithms that tackle this task already exist, but they have different benefits and drawbacks, which leads to a large number of possible reconstruction results (i.e., meshes). The evaluation of those techniques requires extensive comparisons between different meshes which is up to now done by either placing images of rendered meshes side-by-side, or by encoding differences by heat maps. A major drawback of both approaches is that they do not scale well with the number of meshes. This paper introduces a new comparative visual analysis technique for 3D meshes which enables the simultaneous comparison of several meshes and allows for the interactive exploration of their differences. Our approach gives an overview of the differences of the input meshes in a 2D view. By selecting certain areas of interest, the user can switch to a 3D representation and explore the spatial differences in detail. To inspect local variations, we provide a magic lens tool in 3D. The location and size of the lens provide further information on the variations of the reconstructions in the selected area. With our comparative visualization approach, differences between several mesh reconstruction algorithms can be easily localized and inspected.

Index Terms—Visual analysis, comparative visualization, 3D data exploration, focus+context, mesh comparison

---

1 INTRODUCTION

Polygonal meshes are one of the most commonly used surface representations in 3D computer graphics. Their explicit description of the surface location in 3D together with local connectivity information enables memory-efficient storage and provides a convenient data structure for a wide range of applications (e.g., in geometric processing). For many tasks related to mesh creation and/or editing, a multitude of proposed methods exist. Polygonal meshes may serve both as input and output for a majority of such techniques.

As a consequence, the characteristics and capabilities of different approaches for a common task have to be evaluated on the basis of their results, which inevitably leads to the need to compare an – often large – number of *similar* meshes. While formal geometric properties (e.g., polygon areas) can be evaluated by purely mathematical methods, aesthetic considerations almost always require a human in

---

- *Johanna Schmidt, Vienna University of Technology, Austria*
- *Reinhold Preiner, Vienna University of Technology, Austria*
- *Thomas Auzinger, Vienna University of Technology, Austria*
- *Michael Wimmer, Vienna University of Technology, Austria*
- *M. Eduard Gröller, Vienna University of Technology, Vienna, Austria*
- *Stefan Bruckner, University of Bergen, Norway*

the loop. The pleasantness of the final form is of major importance for several geometric processing tasks such as mesh resampling, mesh denoising, and mesh reconstruction from point clouds. Especially the last example is currently a hot topic in research [6], as the advent of affordable scanners has made the creation of virtual representations of real-world objects a commodity. Beyond computer graphics, other fields that deal with 3D objects, like CAD or biomolecular modeling, would also benefit from new trends for multi-mesh comparison.

As an exemplary use case, we further concentrate on *mesh reconstruction*, which refers to extracting meshes from point clouds as accurately as possible. A wide variety of techniques has already been developed, and these algorithms differ (more or less) subtly in their reconstruction behavior – especially in the presence of noise, outliers or other errors in the input data [7]. Furthermore, with almost every technique the output depends on several, partly very sensitive, parameters with varying suitability for different kinds of data. All these facts create a large space of possible results when reconstructing a mesh from a point cloud. Especially the evaluation of a new technique currently requires extensive laborious comparisons, not only between samples of the approach in its own parameter space, but also to existing state-of-the-art methods. Such comparison tasks are additionally complicated by the fact that the desired outcomes can be highly task-dependent and difficult to quantify, which makes visual inspection unavoidable.

Comparative visualization refers to the process of visually depicting differences and similarities in two or more datasets [23]. Within the last years various comparative visualization systems have been developed, which demonstrate that there is a strong demand for the support of comparison tasks in various domains. According to Gleicher et al. [16] there are three main approaches to compare data: juxtaposition (i.e., side-by-side comparison), superposition (i.e., blending), and ex-
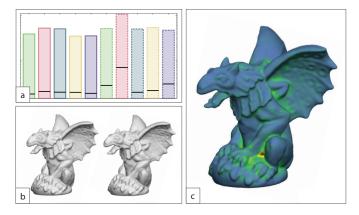
Fig. 2: Common mesh comparison approaches. Current tools employ either statistical evaluation (a), juxtaposition (b) or explicit encoding by color (c) to show differences between meshes (Section 2).

plicit encoding (i.e., difference encoding by some abstract parameter).

Up to now, the tool-set for *visual comparisons of mesh reconstructions* is limited to statistical evaluation (e.g., global error), simple juxtaposition, or explicit encoding by color (Figure 2). The existing approaches do not scale well with the number of instances, and basically support only pairwise comparisons. Color-coding of differences only partially characterizes the behavior of the underlying algorithm (e.g., whether the data is smoothed).

We believe that a combination of explicit encoding, juxtaposition and quantitative measures can support mesh comparison tasks and provide more insight into the underlying data. We propose YMCA, a new comparative visual analysis approach which allows users to compare several meshes against each other. Our application, on the one hand, helps to identify areas in the data where reconstruction algorithms produce different results, and also allows for a detailed exploration of local variations. On the other hand, our system supports users in gaining insight into the characteristics of different mesh reconstruction algorithms. The main features of our approach are:

- Comparison of multiple entities: Our visual analysis methods are designed to overcome the problems of previous approaches that do not scale well with a larger number of meshes. With our approach users are able to get an overview of all studied algorithm results. It is also possible to evaluate the performance of individual algorithms against others.

- Focus+context: As a starting point of the analysis process, we provide an overview of the comparison results. Users can then further concentrate on local variations and explore them in more detail without losing the context information.

- Flexibility: The proposed visual analysis tools can be applied to different mesh comparison tasks, e.g., comparing meshes after mesh simplification, as well as comparing different reconstructed meshes. The approach is neither tied to certain type of mesh (e.g., watertight mesh), nor to a certain mesh comparison metric.

The paper is organized as follows: Section 2 contains a survey of previous work related to the topics of comparative visualization, mesh reconstruction, and focus+context interaction techniques. Section 3 provides an overview of the tasks and challenges we address. In Section 4, the process of identifying mesh differences is described, and our interactive visualization tools are introduced. Implementation details are discussed in Section 5 and results are presented in Section 6. We also collected feedback which is described in Section 7 to evaluate our approach. In Section 8 we discuss the advantages and limitations of our approach and an outlook on future work is given. The paper is concluded in Section 9.

2 RELATED WORK

The work presented in this paper is located in the field of *comparative visualization*. In the last years a great variety of systems and approaches for comparative visualization have already been developed, which are discussed in this section. Since we compare several meshes in 3D, we included a comparison of our approach to existing techniques in the area of *mesh comparison*. The meshes we use have been reconstructed from point clouds, and here we revert to well-studied findings from the field of *surface reconstruction*. Our work makes use of several well-known interaction concepts (i.e., *focus+context* and *linking and brushing*), which are also discussed in this section.

Comparative visualization. Some approaches from the field of comparative visualization deal with comparing 2D data (e.g., images [33]), but there is also a representative group of systems used to compare 3D data. Some of these approaches analyze multiple 3D data structures (e.g., shapes or volumes) by comparing 2D representations of the data [14, 24]. Malik et al. [21] proposed a method to compare different volume datasets by analyzing 2D slices. Other methods concentrate on the comparison of data in 3D, similar to our approach. Masuda et al. [22] visually analyzed 3D shapes of ancient Chinese bronze mirrors by color-coding their differences. Watashiba et al. [31] used critical point graphs to depict similarities in volume datasets. Alabi et al. [1] presented a method for side-by-side comparison of surfaces in 3D. All together, these systems show the demand for tools that explicitly support comparison tasks in 2D as well as in 3D.

Mesh comparison. Due to the need to evaluate mesh editing tools (e.g., for mesh simplification), many approaches have been developed that support mesh comparison. Various techniques focused on the mathematical background and established metrics which can be used to compare meshes. Aspert et al. [4] proposed an approach to measure differences between two meshes by using the Hausdorff distance. Roy et al. [26] introduced a new mesh comparison method using an attribute deviation metric. *MeshLab*, by Cignoni et al. [9], was implemented to combine mesh comparison as well as mesh editing tools. In our work we focus on visual support for mesh comparison, and some interesting approaches have already been developed in this area. Cignoni et al. [11] presented *Metro*, a system that allows for pairwise comparison of surfaces. A similar approach was later proposed by Silva et al. [28]. Their system, which is called *PolyMeCo*, allowed users to compare meshes against a reference mesh. Existing approaches for mesh comparison use color to encode the differences and present the results by juxtaposition. Therefore, they are limited to a small number of meshes. Apart from zooming, the systems also do not provide means to inspect local areas. In our approach we extend these ideas to provide means to compare multiple meshes, and to inspect local regions in more detail.

Surface reconstruction. The acquisition of virtual representations of scanned real-world objects from point clouds is referred to as surface reconstruction. In contrast to point-set surface-representations [2, 3], this paper focuses on mesh reconstruction from point clouds. Meshes are reconstructed according to different formulations of implicit surfaces defined on the input points, ranging from locally fitted tangent planes [18], radial basis functions [8], to Poisson reconstruction [20]. All these techniques exhibit their own characteristic reconstruction behavior in terms of robustness and accuracy, and require various parameters which influence the result. Berger et al. [6] present a benchmark tool for surface reconstruction algorithms, where the user can test different algorithms on different point cloud datasets. When presenting the results, they use juxtaposition where rendered models are placed side by side. This way the complex task of finding relevant differences in the data is shifted to the user.

Linking-and-brushing. The concept of linking-and-brushing is well-known in visualization. It refers to the connection of two or more views in a way that a change to the representation in one view affects the representation in the other one as well [30]. Linking-and-brushing is a very flexible concept that can be applied to many different data representations, like 2D data (e.g., scatter plots [5]) as well as 3D
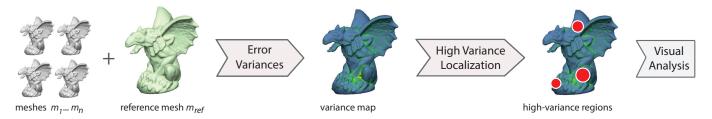
Fig. 3: Overview of our visual analysis approach. The input data consists of a set of n *meshes* and one *reference mesh*. The surface deviations of the meshes are calculated to get the corresponding *variance map* (Section 4.1). Afterwards *high-variance regions* are located in the data (Section 4.2). The results are finally presented in an interactive visualization as described in Section 4.3 and Section 4.4.

data [15]. We use linking-and-brushing to keep track of user selections. Elements in our summary representations can be selected, which will mark them as selected also in the detailed view (and vice versa).

Focus+context. The strength of focus+context and in-place interaction techniques is that they give an overview of the available data, but also allow us to further inspect details on demand. Several focus+context approaches can be found in the literature [12]. Similar to linking-and-brushing, focus+context is a very flexible concept which has already been applied in many different visualization approaches (e.g., scatter plots [25] or sets of images [27]). To locally explore data in 3D, Zhao et al. [32] employed focus+context with local shape preservation by using conformal mapping. Cignoni et al. [10] established a 3D magic lens tool with user-specific content (e.g., different rendering techniques or the result of filtering operations). In our system we use a magic lens tool, which is similar to the two latter approaches. The user can employ this tool to make selections, which will provide more detailed insights into certain data parts. The selection is made on the 3D representation of the data, whereby the context of the selection is always preserved.

3 CHALLENGES OF MULTI-MESH COMPARISON

The need to compare different meshes can be the result of various geometric operations. As an exemplary use case, we focus on the *reconstruction of meshes from point clouds*. Several meshes can be constructed from the same point cloud with different algorithms. The algorithms differ in their reconstruction behavior [7], which means that the resulting meshes exhibit subtle differences. The analysis of such datasets poses interesting challenges. We therefore introduce **Your Mesh Comparison Application (YMCA)** of choice, which addresses the challenges of multi-mesh comparison, as discussed below:

Notion of quality. When reconstructing a point cloud, the user is interested in surfaces that match the shape of the original specimen as accurately as possible. In practice, users often face a trade-off between the preservation of geometric detail and the robust removal of scanning artifacts like noise or holes. From a *statistical* point of view, the quality of a reconstruction is defined by the residual distances of each surface point from the reference shape. However, a statistical evaluation alone hardly communicates a full understanding of a technique's strengths and weaknesses on different types of data. For example, it might happen that algorithms with a low overall error rate smooth certain features in the data, which might not be desired by the user. YMCA presents a general error measurement to the user, and also allows to judge the *visual quality* of the resulting shapes. Giving the user insights into the results and providing him/her the possibility to compare them against each other, also supports understanding the *behavior* of the different reconstruction algorithms. For example, users will identify undesired characteristics like over-smoothing or sensitivity to noise. YMCA provides means to quickly eliminate algorithms from further comparisons if they do not show a desired behavior, which can help to narrow down the search space quite fast.

Complexity and scalability. Current mesh comparison methods provide means to inspect the shape and error of a mesh sample individually [6], or allow for mostly pairwise comparisons among the sample set [9, 28]. These methods quickly become unsuitable for larger sample sets, e.g., when using multiple samples in the parameter spaces of different reconstruction algorithms. YMCA provides a *compact visual overview* that presents individual quality information (e.g., reconstruction error) in the appropriate context and that shows the most relevant differences at a glance. It is no longer necessary to scan/rotate/zoom into several 3D meshes one after another, since the mesh elements can now be explored at once.

Evaluation. For newly developed reconstruction techniques, a common task is their *evaluation* and *classification* with respect to existing methods. So far these required a tedious exploration of the high dimensional space spanned by the input data (e.g., shape, or amount of noise) and the algorithm parameters (e.g., kernel functions or bandwidth). With YMCA it is possible to quickly extract the regions on the mesh where the algorithm of interest shows better/worse results than the other ones it is compared to. YMCA can extract the *most problematic regions* of the mesh, which are those where the reconstruction results have a high variance.

None of the existing methods used for mesh comparison accommodate the above aspects so far (Section 2). YMCA allows for an intuitive, guided and flexible visual analysis of a scalable set of similar meshes to explore the differences among them.

4 YMCA – YOUR MESH COMPARISON APPLICATION

YMCA combines explicit encoding, juxtaposition, parallel coordinates, and interaction techniques (i.e., linking-and-brushing and focus+context) to convey an overview of mesh differences, and to allow the user to inspect local areas of interest.

As mentioned in Section 3, we focus on triangular mesh data produced by different mesh reconstruction algorithms. The data has been created by the surface reconstruction benchmark tool implemented by Berger et al. [6]. The meshes are already registered. No pre-processing (e.g., filtering) has been applied to the meshes. In addition, a reference mesh is assumed to be available for every point cloud. See the discussion in Section 8 for our strategy, if such a reference is not available.

To provide a condensed representation of all differences in the data, we propose to project the variances of the mesh deviations onto the reference mesh. The calculation of the variances is described in Section 4.1. In addition, we locate problematic regions (i.e., regions of high variance) in the model to provide additional guidance to the user when exploring the data. The detection of such regions is outlined in Section 4.2. The problematic regions are used to build a parallel coordinates plot to visualize the performance of the reconstruction algorithms (Section 4.3). The inspection of local areas provides interesting insights into the behavior of the reconstruction algorithms. The visual analysis tools of YMCA are described in Section 4.4. The whole pipeline of YMCA is outlined in Figure 3.

4.1 Calculation of Error Variances

The visual analysis of variations in multiple meshes requires the computation of mesh differences. For the mesh comparison we use an attribute deviation metric as described by Roy et al. [26]. This metric compares meshes in a pairwise manner and calculates point-wise *deviations* from a reference mesh to another mesh. The deviation is
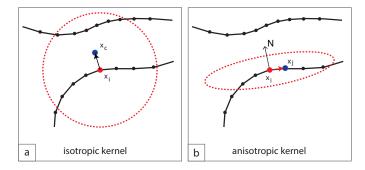
Fig. 4: Surface-aware Mean-Shift. Instead of using an isotropic kernel (a), we employ a surface-aware Mean-Shift with an anisotropic kernel (b) around the current mean x_i. This prevents the means from moving away from the model surface (x_c). Instead, by considering the normal vector N, the means stay close to the intrinsic surface (x_j).

defined as the distance between a vertex of the reference mesh and the nearest point on the surface of the other mesh.

Our approach is not tied to a certain type of metric. In this paper we concentrate primarily on the calculation of the regions of high variance (Section 4.2) and the visual analysis of the mesh differences (Section 4.3 and 4.4). The metric to calculate the mesh differences may be exchanged, as it is outlined in Section 8.

Given n input meshes $M_1, \ldots, M_n$, the mesh comparison results in a set of n surface deviation values (in the following simply denoted by *errors*) for every vertex in the reference mesh M_{ref}. We then calculate the per-vertex error variances (based on the n error values) for all vertices in M_{ref}. In YMCA, we call this discrete distribution of variances the *variance map*.

4.2 Automatic High Variance Localization

To guide the user in the comparative analysis and exploration process, we provide an overview of the surface regions showing the h highest error variances. These are assumed to be the most interesting areas for comparison due to the high disagreement between the reconstruction methods. To this end, we use the variance map computed in the previous section to locate local maxima in the distribution of per-vertex variances on the surface. These maxima are found by employing a weighted Mean-Shift [13] algorithm in $\mathbb{R}^3$ on the set of mesh vertices, where the per-vertex variances define the weights. This is done once in a pre-processing step, directly after the variance map computation.

A set of initially random samples x_i, $i = 1 \ldots N$ of mesh vertices are iteratively shifted toward modes in the variance distribution using a smooth kernel $\theta(r) = e^{-4(r/s)^2}$ of finite support s. One iteration step is given as

$$x_i' = \frac{\sum_j \theta(\delta_{ij}) \sigma_j^2 \, p_j}{\sum_j \theta(\delta_{ij}) \sigma_j^2} \tag{1}$$

where σ_j^2 gives the variance at vertex j, while p_j is its location in $\mathbb{R}^3$ and δ_{ij} denotes its distance to the sample x_i.

However, an isotropic kernel might let samples move away from the intrinsic surface described by the local neighborhood of vertices (Figure 4a). Thus, we need to constrain the sample movements close to the local surface around x_i (Figure 4b). We employ a *surface-aware* distance metric δ_{ij}, which incorporates the surface normal n_i into the weighting kernel as given by

$$\delta_{ij} = \|p_j - x_i\| + |\langle n_i, p_j - x_i \rangle|. \tag{2}$$

After the samples converged to different high-variance modes on the surface, similar points are discarded, and the remaining ones sorted by their amplitude. This gives a list of *hot-spots* that we use in the consecutive visual analysis procedure (Section 4.4). Besides the positions p_i of the hot-spots, we are also interested in their extent, which is given by the weighted sample standard deviation σ of variance values at every hot-spot.

4.3 Parallel Coordinates Plot

The list of hot-spots created in the previous section can be used to span a multi-dimensional feature space. The space is defined by the number of hot-spots h and the error values e_n for every input mesh at the hot-spot positions. We propose to use the high-dimensional visualization technique of *parallel coordinates* [19] to analyze this multi-dimensional feature space. Every hot-spot defines one axis in the parallel coordinates plot, and the dimensions of the axes are given by the global minimum and maximum error values. The input meshes are represented as lines in the plot (Figure 5).

The axes in the plot are initially sorted according to the hot-spots' weighted sample standard deviation σ of variance values. The sorting of the axes can be interactively changed by the user. The parallel coordinates plot gives an overview of the error rate of the reconstruction algorithms in regions of high variance.

4.4 Visual Analysis

YMCA provides interactive tools to explore the results of the mesh comparison analysis. The main elements of the user interface are illustrated in Figure 1.

Overview image. To provide an overview of the differences in the data, we propose a rendering of the reference mesh (Figure 1a). A **heat map** is projected onto the mesh according to the current variance map. The default heat-map colors range from blue (low variance) to red (high variance). If necessary, the color scale can be changed by the user by selecting different colors for the minimum and maximum variance values. The reference mesh rendering allows the user to inspect differences in the data without having to inspect all individual meshes, because the relevant information is aggregated in one view. An example for an overview image can be seen in Figure 6.

Hot-spots and parallel coordinates. In the user interface, the hot-spots are arranged in a parallel coordinates plot as described in Section 4.3. The **parallel coordinates plot**, where the hot-spots are embedded in, represents the input meshes as lines, indicating their local error value at the hot-spot positions (Figure 1b). To interact with the data, the users can change the ordering of the axes and also eliminate individual hot-spots by mouse interaction. It is also possible to create new hot-spots during the analysis (see below). To give the user an idea about the position and size of the hot-spots, they are represented by **thumbnail images** of the mesh displayed at the corresponding parallel coordinates axes (Figure 1c). The thumbnail images are created when a hot-spot is created in the system. The reference mesh is used to generate the images, and the viewports are given by the corresponding hot-spots' locations. The images can be activated by mouse interaction in the interface. When clicking on one of them, the
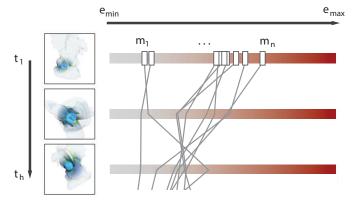

Fig. 5: Parallel coordinates plot. We can use the hot-spots (Section 4.2) to create a multi-dimensional feature space, defined by the number of hot-spots $t_1 \ldots t_h$ and the error values e at those positions for all input meshes. The hot-spots are represented as thumbnail images, and the input meshes $m_1 \ldots m_n$ are defined by lines.
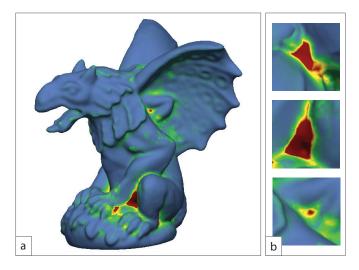
Fig. 6: Overview image. In this figure the reference mesh with a projected heat map according to the variances in the data is shown (a), as well as some mesh regions in more detail (b).

overview is automatically rotated to the location of the hot-spot. The optimal viewing angle is calculated by aligning the viewing direction with the normal vector of the hot-spot and centering it in the viewport. To emphasize the hot-spot locations, we use a focus+context approach in the rendering of the mesh. We employ opaque rendering only to the hot-spot area, while the rest of the mesh is depicted with high translucency (Section 4.5). The user can use the parallel coordinates plot and the hot-spots to quickly depict input meshes containing undesired results and eliminate them from further analysis.

Magic lens. Besides providing an overview of the data, YMCA also supports means to further inspect local variations. We propose a **magic lens** tool (Figure 1d) which can be used to select a certain region of the mesh. The magic lens is circular, because drawing a circle is a very intuitive way of selecting objects. A colored circle drawn over the mesh indicates the current location of the lens. The size can be dynamically adjusted. Since the circle is transparent, the selected part of the reference mesh is also shown.

Analyzing local variations. The current location and size of the magic lens tool is used to present more detailed information about the local behavior of the mesh reconstruction algorithms in a **detail view** (Figure 1e). To provide quantitative information, a **ranking** of the mesh algorithms is provided. The reconstruction algorithms are sorted according to their corresponding accumulated error at the current lens position. For every algorithm a rectangle is placed at the corresponding position along an error scale. The scale ranges from the global minimum to the global maximum error. The user can activate rectangles by mouse interaction to reveal the name of the algorithm at this position. In addition to the ranking, further visual information is needed for the analysis of the meshes, for which we added a **data summarization** view. Here the meshes are *classified* according to their accumulated error at the current lens position (Section 4.6 for further information). This summary gives an overview of the variance and possible problems at the current lens position. According to feedback from domain experts, besides having an overview, it still is necessary to have access to the individual meshes. Therefore, we allow the user to see **close-up views** of the reconstructed meshes. If more than one mesh is selected, we place the close-up views side-by-side according to the concept of Small Multiples [29]. The meshes inside the close-up views are color-coded according to the accumulated error at the current lens position. Here we use a different heat map than the one projected onto the reference mesh to make a clear distinction between the variance and the local error values. All interface items are updated every time the magic lens is moved or resized. An example of how a detail view may look like can be seen in Figure 7.

Additional user controls. We provide some additional controls which can be used to adapt the system's interface elements according to individual preferences. As mentioned above, the color scales of the overview image and the close-up views of the reconstructed meshes can be customized by the user. In addition, the upper and lower bound of the color heat map in the overview image can also be adjusted, which allows the user to concentrate on different variance ranges. The render mode can be changed from hot-spot rendering to heat-map rendering at any time. In the detail view, the user can decide whether he/she wants to see the global ranking as well, and whether he/she wants to concentrate on the data summarization, on the individual meshes, or both. Furthermore, it is possible to replace the reference mesh by some other input mesh. Then all differences and variances are re-computed and the data is re-loaded. This option allows the user to compare one mesh against all others in the dataset.

4.5 Hot-Spot Rendering

The hot-spots in YMCA can be dynamically activated in the user interface by selecting a hot-spot thumbnail in the parallel coordinates view (Section 4.3). This automatically rotates the overview to an optimal viewing angle to uncover the region of interest on the reference mesh. However, in many cases the hot-spots are located in concavities of the surface, which are often occluded by other parts of the mesh. Thus, a clear view onto the hot-spot may be prevented, and the user might lose the focus if he/she rotates the model.

We therefore use a visualization technique that removes any occlusions of the interesting surface region by increasing the transparency with the distance to the hot-spot. Given a pre-computed hot-spot position p and its extent σ, we employ a smooth transparency kernel $K(x) = e^{-4\|x-p\|^4/\sigma^4}$ to put the hot-spot into focus (full opacity) while removing occluding surface parts and at the same time providing background context (high transparency).

This is done by ray casting, using two render passes: First, in an accumulation pass, the whole mesh is drawn into a texture using accumulative blending. Every fragment with corresponding surface position x is weighted by the kernel $K(x)$. This way, the resulting texture stores for each pixel the weighted sum of surface colors along the respective ray, as well as the sum of all weights. Then, in the normalization pass, the accumulated values in the texture are normalized by the sum of their weights and drawn onto the screen (Figure 8).
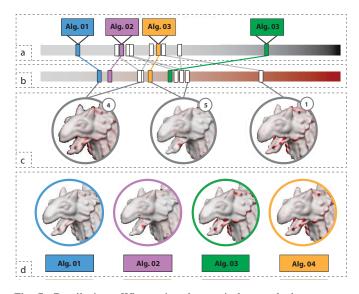

Fig. 7: Detail view. When using the magic lens tool, the user can hover over the reference mesh and inspect parts of it in more detail. The global error value of the input meshes (a) and the local error at the lens position (b) are displayed at the top. The meshes are classified according to their local error value at the current position of the lens (c). It is also possible to view individual mesh renderings (d).
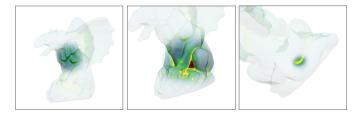
Fig. 8: Hot-spot exploration. This figure shows a model rendered in hot-spot mode (Section 4.5) with three different hot-spot examples (different position and extend).

4.6 Data Summarization in the Detail View

To deal with scalability, and to provide a condensed overview of the data, we decided to integrate a data summarization into the detail view (Figure 7c). When hovering with the magic lens over the mesh, the user should details about the underlying data, as well as a combined summary. The purpose of the summary is to quickly inform the user about the variability of the data at the current lens position, and indicate whether further inspection would be necessary.

We propose to classify the reconstructions according to their accumulated error at the current lens position. The variance at the current lens position indicates the number of classes which are needed. After detailed discussions with our collaborators we came to the conclusion that dividing the data into three classes (best/middle/worst) is a very intuitive way of presenting a summary of the data. Therefore, a maximum of three classes is allowed. We use two fixed thresholds that define the final number of classes. An average image is used as class representative to display the data.

Figure 9 gives an example of how the data summarization could look like. If the variance at the current lens position is low, only one class is created containing all meshes (Figure 9a). This shows that all reconstruction algorithms produced the same result at this position. The higher the variance, the more classes are created (Figure 9b). Such positions on the mesh, where the reconstruction algorithms produced very different results, might need further inspection by the user.

The proposed data summarization provides a good overview of the data, where the user can quickly decide about a further local inspection of the data. In addition to this level of abstraction, the average images representing the classes still reveal the underlying information.
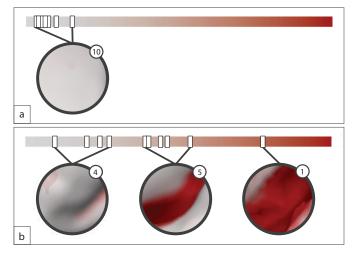

Fig. 9: Data summarization. This figure shows two examples for data summarization in the detail view when hovering over areas of low (a) and high (b) variance.

5 IMPLEMENTATION

The pre-processing step, consisting of comparing the meshes and calculating the variances, was implemented in C++. We used *MeshDev* [26] to calculate the differences. The cost of the pre-processing step, consisting of the calculation of the variance map and the location of the hot-spots, depends on the number of meshes in the input data set. No user input is required during the pre-processing step. The interaction itself, which works in real-time, has been implemented in C++ and OpenGL/GLSL. The application was tested on a machine with an Intel i7 CPU, 12 GB of RAM and an NVIDIA GeForce GTX 580 graphics card. A comparison of the computation times and memory requirements during the analysis can be found in Table 1. A more detailed description of the datasets can be found in Section 6.

Table 1: Runtime and memory requirements. The first column gives the name of the dataset. The second column shows the number of meshes in this dataset. The next column contains the runtime for the pre-processing, which consists of the calculation of the variances and the localization of the hot-spots. The last column shows the amount of memory used on the graphics card.

Dataset	Meshes	Pre-Processing	Memory
Gargoyle	10	12.7s	82.6mb
Dancing-Children	100	86.9s	797.5mb
Daratech	15	17.2s	131.4mb

6 RESULTS

We used data from the field of point-cloud reconstruction to test our approach (see also Section 4). The data was produced by different algorithms, for example Poisson Surface Reconstruction (*Poisson*), Algebraic Point Set Surfaces (*APSS*) and Multi-level Partition of Unity Implicits (*MPU*). The reader is referred to the survey by Berger et al. [6] for a more detailed description of the reconstruction algorithms.

We applied our approach to three different datasets. The first dataset, called *Gargoyle*, comprises ten mesh reconstructions from a virtual point cloud scan of a carved stone figure. The second dataset, called *Dancing-Children*, consists of 100 mesh reconstructions from a virtual point cloud scan of an ornament. The third dataset, called *Daratech*, comprises 15 mesh reconstructions from a scan of an industrial workpiece. Figure 10 shows renderings of the reference meshes of the three datasets and further information about the mesh dimensions.

Different reconstruction algorithms perform with different accuracy on different parts of a surface. The parallel coordinates plot in conjunction with the hot-spot thumbnails (Section 4.3) enables the user to understand the relative performance of different algorithms on a particular part of a surface, and at the same time allows him/her to visually inspect the reconstructed surface and its quality. In Figure 11a, YMCA visualizes an automatically identified artifact in the *Daratech* dataset produced by the *Wavelet* algorithm (wrong mesh vertices highlighted red), and clearly classifies this algorithm as an individual outlier on the
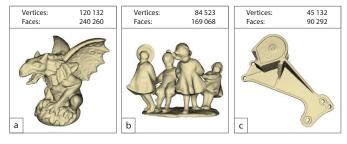

Fig. 10: Datasets used to evaluate YMCA. All datasets consist of mesh reconstructions from point cloud scans. The first one (a) is called *Gargoyle*, the second dataset (b) is called *Dancing-Children*, and the name of the third dataset (c) is *Daratech*.
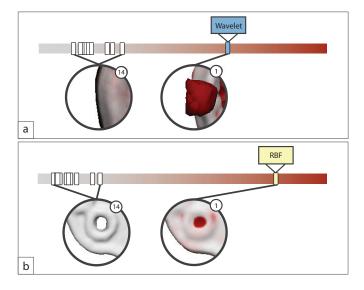
Fig. 11: Outlier detection. The visual analysis tools of YMCA allow users to quickly detect outliers, which might be caused by noise in the data (a) or by a certain algorithm behavior like over-smoothing (b).
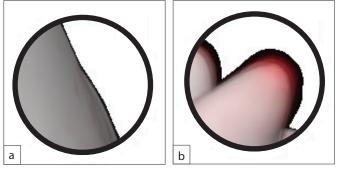

Fig. 13: Mesh boundaries. The summary view in the detail view enables users to identify regions where the reconstruction algorithms produce almost the same (a) or different (b) mesh boundaries, indicated by a color band.

parallel coordinate axis, with relatively high reconstruction error. At a different part of the model, the *RBF* algorithm stands out by falsely closing a hole, where all remaining algorithms perform correctly (Figure 11b). By giving this integrated overview of the algorithms' relative performance on different surfaces areas (statistically and visually), the hot-spot localization and the parallel coordinates plot allow the users to quickly classify algorithms and judge their eligibility. A manual comparison of all meshes is far more tedious and leads to particular artifacts being easily missed.

In all three datasets, parts of the variance map exhibit rectangular-shaped artifacts. We used the overview image and the magic lens (Section 4.4) to further inspect those areas. With our tools we could find out that these artifacts are always produced by the Poisson reconstruction algorithm (Figure 12). Apparently, this artifact is caused by the limited resolution of the octree employed by the Poisson algorithm for reconstruction. With the visual analysis tools of YMCA, this artifact could quickly be related to the Poisson algorithm and explored visually. It is clearly visible in which part of the model the octree resolution has to be adjusted to guarantee a smooth reconstruction.

With the data summarization used in the detail view (Section 4.6), differences at the reconstructed mesh boundaries can be explored. Blending the lens view of meshes of the same class allows a direct comparison and visualization of the geometric variance of their boundaries. In Figure 13, two examples of different boundaries can be seen.

Parts of the boundary of the *Gargoyle* model have been reconstructed in a similar way by all algorithms (Figure 13a), whereas in other parts the boundaries of the reconstructed meshes differ more strongly (Figure 13b). The differences in the boundaries become visible as a color band in the images. Different reconstructions may produce different boundaries, due to different approaches to fit a surface to the point cloud data. With YMCA it is now possible to explore these effects in detail. The regions where boundaries differ are clearly visible when inspecting the mesh with the magic lens. This helps the user to judge which reconstruction would better represent the data.

The parallel coordinates plot of YMCA also proved to be very helpful in the analysis of large datasets. If such datasets shall be inspected, it is very important to quickly narrow down the search space to suitable reconstructions. Reconstruction algorithms that perform either generally very good, or very bad, in most cases can be identified. It is also possible to eliminate algorithms from further analysis, if they do not meet certain reconstruction requirements. An example is given by the inspection of the *Dancing-Children* dataset, consisting of 100 meshes. The resulting parallel coordinates plot can be seen in Figure 14. Here it was possible to, for example, identify one algorithms (*Fourier-3*) with a low error rate, and one algorithm with a high error rate (*SPSS-7*) in all hot-spot regions. We also found one algorithm (*Scattered-2*) that exhibits a varying error rate.
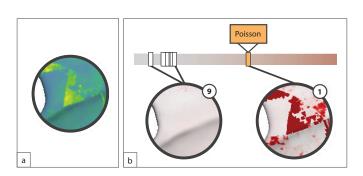

Fig. 12: Artifact analysis. In all datasets rectangular-shaped artifacts could be identified in the overview image (a). With the magic lens tool it is possible to find out that these artifacts are caused by the *Poisson* reconstruction algorithm (b).
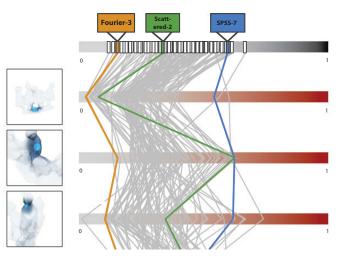

Fig. 14: Analyzing large datasets. The parallel coordinates plot is a very helpful tool when analyzing large datasets. One algorithm with a low error rate (*Fourier-3*), one with a high error rate (*SPSS-7*) and one with a varying error rate (*Scattered-2*) in the hot-spot regions could be identified very quickly.

7 EVALUATION

To evaluate our approach, we have collected qualitative feedback from users experienced in working with meshes. From this feedback we wanted to find out how useful the proposed visual analysis is for the users, and we also discussed possible applications for future work.

Every feedback session lasted between 30 and 45 minutes. First the motivation and the technique itself were explained to every participant. Then they got a training dataset where they could test the interaction possibilities. Afterwards they were presented a new dataset, and were asked to name one or more reconstruction algorithms that produce appropriate results for the given point cloud. They were also asked to explain their decision. For this task they had ten minutes time. At the end, we asked them to fill in a questionnaire with four questions (described in Figure 15 and Figure 16).

We asked seven participants (six men, one woman) to participate in our feedback study. Three of the participants have been working in the field of point-cloud reconstruction for years, and therefore have a lot of experience. Two other participants are experienced computer scientists in the rendering field, where they are working with mesh operations like filtering or simplification. Two participants are students from the field of computer graphics. All participants agreed that analyzing point cloud reconstructions is an important task, and that existing methods do not provide sufficient assistance for this.

Six out of seven participants had no problem to solve the task of finding an appropriate reconstruction algorithm for the given point cloud. One participant ran out of time while solving the task. We compared the results with reconstructions that were previously selected by domain experts. It turned out that participants selected the most suitable reconstruction algorithms in all cases.

With the first three questions in the questionnaire we wanted to find out more about the practicability of the system:

1. Does the system help to spot point cloud regions which are problematic for reconstruction?

2. Does the system help to decide which reconstruction algorithm should be used?

3. Does the system help to better understand the strengths and weaknesses of certain reconstruction algorithms?

The answers to these three questions can be seen in Figure 15. The participants agreed that YMCA helps to spot the most problematic regions in the reconstruction from a point cloud (*Question 1*). They also largely agreed that the system can help to identify the most appropriate reconstruction algorithms for a given point cloud (*Question 2*). However, they were discordant about whether YMCA helps to better understand how the reconstruction algorithms work (*Question 3*). Some participants stated that algorithms can be judged in a visual way, but for a detailed analysis additional information about the point cloud (i.e., noise level) would be necessary. For future work, it would be interesting to analyze the algorithms' pipelines in more detail, and to also take into account the influence of different parameter settings.

The fourth question in the questionnaire concerned which elements of the user interface the participants found helpful during the analysis. We asked which of the following elements they used the most:

- Variance map (i.e., overview image)
- Parallel coordinates (i.e., visualization of reconstruction results in the parallel coordinates plot)
- Hot-spot localization (i.e., the possibility to click on hot-spot thumbnails in the parallel coordinates plot and the hot-spot rendering mode)
- Detail view and data summarization (i.e., detail view with close-up views and ranking, and data summarization)

The answers to this question can be found in Figure 16. The overview image showing the variance map was rated to be the most helpful interface element for the users. This is not surprising, since at the one hand this is the central interaction element of the system, and on the other hand most users are already familiar with interpreting color heat maps on 3D models. The parallel coordinates plot was very helpful for the participants to compare the overall and local performance of individual algorithms. They used this interface element especially to eliminate reconstructions from further analysis. Although all users were convinced about the fact that a list of hot-spots is already prepared when starting the analysis, some of them did not like the hot-spot rendering technique. They stated that it is confusing at the beginning and needs more experience to be interpreted in the right way. The participants also used the detail view to judge the local behavior of the algorithms. Only one participant stated that the data summarization is sometimes hard to interpret and would need a longer training period.

We also asked the participants about suggestions for future work. During these discussions it turned out that the system inspired the users quite a lot, and they had many suggestions for additional features and applications. For the overview image, one participant stated that it
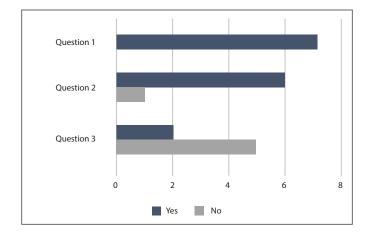

Fig. 15: Evaluating the practicability of YMCA. *Question 1:* Does the system help to spot point cloud regions which are problematic for reconstruction? *Question 2:* Does the system help to decide which reconstruction algorithm should be used? *Question 3:* Does the system help to better understand the strengths and weaknesses of certain reconstruction algorithms? YMCA proved to be very helpful for evaluating point clouds and finding the most suitable reconstruction.
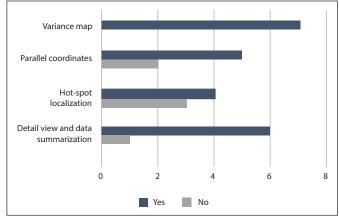

Fig. 16: Evaluating the most helpful user interface elements. With this question we wanted to find out which user interface elements the users found the most helpful ones. All participants liked the variance map and rated it to be very useful. Also the parallel coordinates plot and the detail view were used by most of the participants. However, the hot-spot rendering was sometimes confusing and therefore not rated to be very useful in all cases.

would be helpful to see the reference mesh rendered in colors according to the algorithms which perform best at certain points. This could be a valuable hint in the analysis. Having a very colorful model means that algorithms differ a lot, whereas having large uniformly colored parts means that one algorithm performs better than all others in those areas. For the data summarization in the detail view, the participants suggested that sometimes it would be useful to be able to manually adjust the class borders, or even do a classification by themselves (by manually selecting algorithms).

The participants experienced with point cloud reconstruction also stated that they liked the data summarization in the detail view, because it quickly provides an overview of the data at the current local position. They also pointed out that summarization alone is not helpful for them. To be able to judge which algorithm performs better than the others, they still need to be able to access the individual input meshes. Therefore, they also liked the possibility to depict all close-up views of all meshes from one class in a Small Multiples display.

8 DISCUSSION

YMCA operates on a set of meshes which are compared to a reference mesh and analyzes similarities and differences among them. Not many suitable tool-sets exist that allow the efficient comparison of multiple meshes. With YMCA it is now possible to quickly and visually analyze mesh reconstruction results and depict the appropriate solution for the given point cloud data. The system also provides an overview of the critical areas in the available point cloud data, with respect to different reconstruction approaches. Our system nonetheless has some limitations, which point to interesting directions for future work.

For the data in this paper, reference meshes were given which could be used to calculate the differences in the meshes. However, often this is not the case when analyzing point cloud data (e.g., when scanning real-world objects). In this case we propose to create an average mesh out of the input meshes, and to compare the meshes against this average. This provides an initial overview of the differences in the data. If the user is not satisfied with comparing the meshes against the average, he/she can exchange the reference and use some other input mesh instead, e.g., the mesh that delivers the best reconstruction of certain parts of the input data. YMCA already provides controls for exchanging the reference mesh.

We used an attribute deviation metric [26] to calculate the mesh differences. This paper is primarily about the visual depiction of mesh differences, which means that the calculation of the differences is decoupled from the visual representation. The metric can be exchanged with any other mesh difference calculation (e.g., curvature measurements). We tested this by using geometric deviation [26], which can be seen in Figure 17.

The variance map of YMCA gives an overview of the regions in the point cloud where the reconstructions produce different results. However, for some applications it might be interesting to see the global error instead (i.e., regions where all reconstructions fail). To test this, we used the mean squared error to aggregate the different errors into one value per vertex. YMCA already provides means for exchanging the metric, which can be also applied to using the error values for

visualization instead of the variances. At present this means that the system has to be initialized with either the one or the other settings. The user can only explore the results of one metric at a time. This is something that we want to change in the future. We would also like to work on possibilities where the metric can be changed during analysis, while the settings (like selections in the parallel coordinates plot) are still preserved for all metrics. This way it will be possible to select a list of hot-spots (calculated by different metrics) and use them for further analysis.

As pointed out during the evaluation, YMCA provides limited support in understanding the strengths and weaknesses of individual reconstruction algorithms. Our approach enables the user to visually compare the results and therefore judge them, but domain experts stated that additional information about the input data (e.g., noise level) would be helpful. We plan to integrate this into the system in the future.

Berger et al. [6] implemented a surface reconstruction benchmark tool which can be used to test several reconstruction algorithms against one point cloud dataset. Additionally, the benchmark tool provides scanning simulations which can produce different point clouds of the same model with different quality (e.g., add more or less noise to the data). Then the benchmark tool can be used to apply the reconstruction algorithms to different point cloud versions. This would span a parameter space for every reconstruction algorithm showing its strengths and weaknesses (e.g., in the presence of noise).

We also would like to apply YMCA to other mesh datasets, for example created by different mesh re-sampling or simplification algorithms. It would also be interesting to explore other 3D datasets. One possibility could be to explore differences in algorithms for iso-surface construction. In the user interface we would like to add more complex shading models with the possibility to change the lighting by user-defined parameters. For the analysis we would like to provide means that the user can define a new reference (e.g., a specific corner or shape) the input meshes should be compared to.

During the evaluation, the domain experts also brought up another promising idea for future work. For them it would be very helpful to be able to export a 2D flattened or unrolled representation of the current model showing all, or at least the most important, hot-spots in one image, together with samples of the reconstructed meshes. Up to now such illustrations are generated manually by rotating the model and producing close-up views.

9 CONCLUSION

In this paper we presented *Your Mesh Comparison Application* (YMCA), a new visual analysis application for the comparative visualization of multiple 3D meshes. Interactive tools are provided to present an overview of the differences in the data, and to explore local areas of interest in more detail. Our visualization approach combines explicit encoding, juxtaposition, and parallel coordinates. It further addresses the scalability problem of previous mesh comparison tools.

We applied our approach to meshes coming from different mesh reconstruction algorithms that were applied to point clouds. With our method, differences between several resulting meshes can be quickly identified, and it is also helpful to explore individual characteristics of the different mesh reconstruction algorithms.

In the future, it will be interesting to investigate other possible interaction techniques. Especially for large datasets, we would like to improve the interaction possibilities in the parallel coordinates plot (e.g., by angular brushing [17]). We would also like to combine several mesh comparison metrics into one system. Another direction for further research is the fine-grained analysis of the parameter space of the available mesh reconstruction algorithms. With our tool it could be possible to identify strengths and weaknesses of the algorithms when applied to different point clouds.
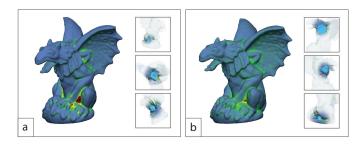

Fig. 17: Changing the metric from point distances (a) to geometric deviations (b) results in an alternative overview image as well as different hot-spot locations.

REFERENCES

[1] O. S. Alabi, X. Wu, J. M. Harter, M. Phadke, L. Pinto, H. Petersen, S. Bass, M. Keifer, S. Zhong, C. G. Healey, and R. M. Taylor. Comparative Visualization of Ensembles Using Ensemble Surface Slicing. *Visualization and Data Analytics*, 8294:1–12, Jan. 2012.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and Rendering Point Set Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, Jan. 2003.

[3] N. Amenta and Y. J. K. Kil. Defining Point-Set Surfaces. *ACM Trans. Graph.*, 23(3):264–270, Aug. 2004.

[4] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. MESH: Measuring Error between Surfaces using the Hausdorff Distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume 1 of *ICME '02*, pages 705–708, Lausanne, Switzerland, Aug. 2002. IEEE.

[5] R. A. Becker and W. S. Cleveland. Brushing Scatterplots. *Technometrics*, 29(2):127–142, May 1987.

[6] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva. A Benchmark for Surface Reconstruction. *ACM Transactions on Graphics*, 32(2):20:1–20:17, Apr. 2013.

[7] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva. State of the Art in Surface Reconstruction from Point Clouds. In S. Lefebvre and M. Spagnuolo, editors, *Proceedings of the Eurographics 2014, Eurographics STARs*, EG '14, pages 161–185, Strasbourg, France, 2014. Eurographics Association.

[8] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 67–76, Los Angeles, CA, USA, 2001. ACM.

[9] P. Cignoni, M. Corsini, and G. Ranzuglia. MeshLab: an Open-Source 3D Mesh Processing System. *ERCIM News*, 2008(73), 2008.

[10] P. Cignoni, C. Montani, and R. Scopigno. Magicsphere: an insight tool for 3D data visualization. *Compututer Graphics Forum*, 13(3):317328, 1994.

[11] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring Error on Simplified Surfaces. *Compututer Graphics Forum*, 17(2):167–174, 1998.

[12] A. Cockburn, A. Karlson, and B. B. Bederson. A review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, Jan. 2008.

[13] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002.

[14] M. DaSilva, S. Zhang, C. Demiralp, and D. H. Laidlaw. Visualizing the Differences between Diffusion Tensor Volume Images. In *Proceedings of the ISMRM Workshop on Diffusion MRI: What Can We Measure?*, ISMRM '02, pages 237–238, St. Malo, France, Mar. 10-12 2002.

[15] H. Doleisch and H. Hauser. Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. In *Proceedings of the Winter School of Computer Graphics*, WSCG '02, pages 147–154, Plzen-Bory, Czech Republic, Feb. 2002.

[16] 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, Oct. 2011.

[17] H. Hauser, F. Ledermann, and H. Doleisch. Angular Brushing of Extended Parallel Coordinates. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '02, pages 127–, Boston, MA, USA, Oct. 2002. IEEE Computer Society.

[18] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '92, pages 71–78, Chicago, IL, USA, 1992. ACM.

[19] A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In *Proceedings of the 1st Conference on Visualization*, VIS '90, pages 361–378, San Francisco, CA, USA, 1990. IEEE Computer Society.

[20] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson Surface Reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Cagliari, Sardinia, Italy, 2006. Eurographics Association.

[21] M. M. Malik, C. Heinzl, and M. E. Gröller. Comparative Visualization for Parameter Studies of Dataset Series. *IEEE Transactions on Visualization and Computer Graphics*, 16:829–840, Sep. 2010.

[22] T. Masuda, S. Imazu, S. Auethavekiat, T. Furuya, K. Kawakami, and K. Ikeuchi. Shape difference Visualization for ancient bronze Mirrors through 3D range images. *Journal of Visualization and Computer Animation*, 14(4):183–196, 2003.

[23] H.-G. Pagendarm and F. H. Post. Comparative Visualization: Approaches and Examples. In *In: Visualization in scientific computing (H. Göbel and H. Müller and B. Urban)*, pages 95–108. Springer-Verlag, Vienna, Austria, 1995.

[24] A. Pang and A. Freeman. Methods for comparing 3D Surface Attributes. *Visual Data Exploration and Analysis III*, 2656(1):58–64, 1996.

[25] H. Piringer, R. Kosara, and H. Hauser. Interactive Focus+Context Visualization with linked 2D/3D Scatterplots. In *Proceedings of the 2nd International Conference on Coordinated & Multiple Views in Exploratory Visualization*, CMV '04, pages 49–60. IEEE Computer Society, 2004.

[26] M. Roy, S. Foufou, and F. Truchetet. Mesh Comparison using Attribute Deviation Metric. *International Journal of Image and Graphics*, 4(1):127–140, Jan. 2004.

[27] J. Schmidt, M. E. Gröller, and S. Bruckner. VAICo: Visual Analysis for Image Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, Dec. 2013.

[28] S. Silva, J. Madeira, and B. S. Santos. PolyMeCo - A Polygonal Mesh Comparison Tool. In *Proceedings of the 9th International Conference on Information Visualisation*, IV '05, pages 842–847, London, UK, Jul. 2005. IEEE Computer Society.

[29] E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.

[30] M. O. Ward. Linking and Brushing. In *Encyclopedia of Database Systems*, pages 1623–1626. Springer US, 2009.

[31] Y. Watashiba, K. Sakai, K. Koyamada, and M. Kanazawa. Visualizing Similarities between Volume Datasets by Using Critical Point Graph and Character Recognition Technique. *The Journal of the Society for Art and Science*, 5(1):11–17, Jan. 2006.

[32] X. Zhao, W. Zeng, X. D. Gu, A. E. Kaufman, W. Xu, and K. Mueller. Conformal Magnifier: A Focus+Context Technique with Local Shape Preservation. *IEEE Transactions on Visualization and Computer Graphics*, 18(11):1928–1941, Feb. 2012.

[33] H. Zhou, M. Chen, and M. F. Webster. Comparative evaluation of visualization and experimental results using image comparison metrics. In *Proceedings of the Conference on Visualization*, VIS '02, pages 315–322, Boston, MA, USA, Oct. 27 - Nov. 1 2002. IEEE Computer Society.