VAICo: Visual Analysis for Image Comparison

Johanna Schmidt, M. Eduard Gröller, *Member, IEEE Computer Society*, and
Stefan Bruckner, *Member, IEEE Computer Society*

Abstract—Scientists, engineers, and analysts are confronted with ever larger and more complex sets of data, whose analysis poses special challenges. In many situations it is necessary to compare two or more datasets. Hence there is a need for comparative visualization tools to help analyze differences or similarities among datasets. In this paper an approach for comparative visualization for sets of images is presented. Well-established techniques for comparing images frequently place them side-by-side. A major drawback of such approaches is that they do not scale well. Other image comparison methods encode differences in images by abstract parameters like color. In this case information about the underlying image data gets lost. This paper introduces a new method for visualizing differences and similarities in large sets of images which preserves contextual information, but also allows the detailed analysis of subtle variations. Our approach identifies local changes and applies cluster analysis techniques to embed them in a hierarchy. The results of this process are then presented in an interactive web application which allows users to rapidly explore the space of differences and drill-down on particular features. We demonstrate the flexibility of our approach by applying it to multiple distinct domains.

Index Terms—Comparative visualization, focus+context visualization, image set comparison

1 Introduction

A common task in data analysis is to compare two or more related results to discover their differences or similarities. In addition to choosing the appropriate similarity measure, it is also important to make use of appropriate techniques to visualize the comparison results. *Comparative visualization* refers to the process of visually depicting differences and similarities when comparing multiple datasets [26].

In the past few years an increasing number of comparative visualization systems have been developed. All together, these systems demonstrate that there is a strong demand for comparison tasks in various different domains caused by the ever-growing amount of acquired data. For example, in biology additional information about certain species can be gained by comparing multiple genetic sequences. In the medical domain, deviations in MRI or sonographic images can indicate anomalies which should be further inspected. In image processing, results of different edge detection or segmentation algorithms have to be compared. In visualization and rendering, results have to be compared against each other to evaluate variances that are caused by different parameter settings.

Comparative visualization tools have to meet certain requirements to be appropriate for a variety of different datasets. One important issue is scalability pertaining to dataset size (i.e., number of items in the dataset). In the case of 2D image comparison, many approaches place the images to be compared side-by-side or in multiple views, or they overlay images semi-transparently. However, due to limitations in the human perceptual capacity as well as due to limited screen space, such comparative visualizations do not scale well. These tools are only suitable for comparing a limited number of images. This is a problem, for example, for the investigation of biological data, where datasets often are based on the analysis of several hundreds of specimens.

Another important issue when developing comparative visualization techniques is how to provide information about the underlying original data. In many approaches differences between datasets are mapped to visual attributes such as colored patterns (Figure 1). Although this clearly highlights differences and similarities between the datasets, it hides the original data that has been used for calculation. Having knowledge about the original data allows us to identify patterns in the datasets (i.e., to detect outliers).

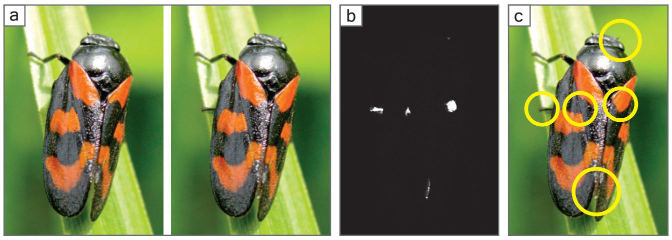

Fig. 1. Image Comparison. This figure shows two pictures that look similar, but in fact exhibit local changes. A very common way to compare them is by placing them side-by-side (a). To help the user to find the variations more quickly, a difference image (b) can be computed. In this illustration information about the similar parts of the data gets lost. Another possibility is to highlight differences by certain patterns (e.g., colored circles) as shown in (c). In this case similar parts of the data are still visible; however, no further information is provided on how the differences are structured.

We believe that an interactive visualization providing insight into the underlying raw data can lead to a better overall understanding of the studied datasets. We propose a new comparative visualization approach which preserves contextual information while allowing us a detailed analysis of the variations between datasets. Our approach provides effective means for examining local differences in a large image dataset. It supports users in gaining a better overview of different image characteristics and allows them to further investigate individual local differences. The main features of our approach are:

- Scalability: Unlike previous approaches, the proposed visualization technique is specifically designed to compare large sets of images (i.e., hundreds of images in one set).
- Focus+Context [5]: Our comparative visualization approach provides an overview of the image differences (i.e., how much of the image space is affected) and allows users to drill-down on individual features.
- Flexibility: Our approach is not targeted to a certain type of image and is not tied to any particular image comparison metric.

- *Johanna Schmidt, Vienna University of Technology, Austria. E-mail: jschmidt@cg.tuwien.ac.at.*
- *M. Eduard Gröller, Vienna University of Technology, Austria. E-mail: groeller@cg.tuwien.ac.at.*
- *Stefan Bruckner, University of Bergen, Norway. E-mail: stefan.bruckner@uib.no.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/13/$31.00 © 2013 IEEE   Published by the IEEE Computer Society

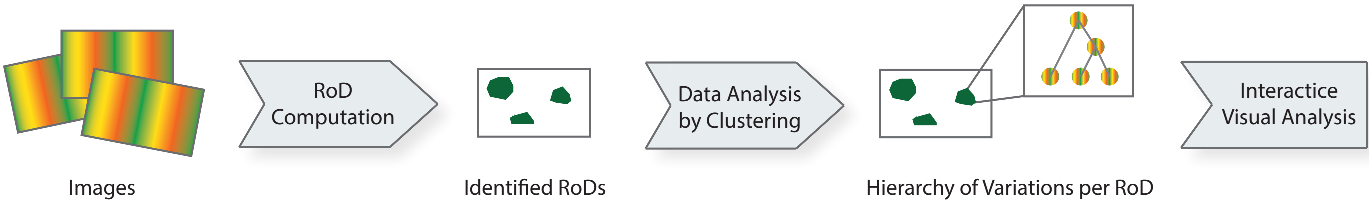

Fig. 2. Overview of our comparative image visualization approach. The input data consists of a set of images which contain small local variations when compared to each other. To locate regions of differences (RoDs), an image comparison metric is applied (Section 3.1.1). In the next step hierarchical clustering is performed on the corresponding image data of every RoD. This classifies the differences according to their significance (Section 3.1.2). The results are then presented in an interactive comparative visualization which is described in Section 3.2.

The paper is organized as follows: Section 2 contains a survey of previous work related to the topic of comparative visualization, parameter studies, focus+context techniques, and hierarchical clustering. In Section 3 an overview of our visualization method is provided. The process of identifying and structuring differences in a set of images is described and the interactive visualization tools are introduced. The implementation details are discussed in Section 4 and results are presented in Section 5. We conducted a user study to evaluate our technique, which is discussed in Section 6. Advantages and limitations are outlined in Section 7. The paper is concluded in Section 8.

2 Related Work

In the last years a great variety of systems and approaches have been developed in the field of **comparative visualization**. For analyzing different light intensities in renderings, Pang and Freeman [27] used color and other parameters like textures to highlight differences. Verma and Pang [38] implemented an approach for comparative flow visualization. To compare video data, Gareth and Chen [13] introduced an approach to visualize differences between video frames in 3D space. They treated a video sequence as a 3D volume and applied volume rendering to it. For analyzing biological data, Munzner et al. [25] provided a visualization technique for comparing large phylogenic trees, and Procter et al. [30] used comparative visualization to check multiple genetic sequences. In archeology objects are often classified by comparison. Masuda et al. [21] visually analyzed ancient Chinese bronze mirrors and their shape differences. Baudrier and Riffaud [1] proposed an approach for comparing ancient documents. For dealing with other data like office documents, Drucker et al. [8] presented a method to spot differences between multiple versions of a PowerPoint presentation. Piringer et al. [29] implemented an interactive approach for a comparative visual analysis of 2D function ensembles. Tory et al. [36] developed new techniques to visualize projects in construction management. All together, these systems show the demand for developing tools that explicitly support comparison tasks. However, the presented approaches are all targeted to a specific domain and to a certain type of dataset. Eler et al. [11] proposed a method to visually analyze image collections. Their visualization method allows feature-based grouping and classification of images, but does not provide means to further inspect individual features.

Existing approaches for **2D comparative image visualization** generally place the objects to be compared side-by-side or in multiple views [37]. Other approaches for 2D image comparison depict images to be compared in the same space. The simplest solution for this is blending (i.e., to overlay images semi-transparently). Kammerer et al. [17] used this method to spot differences between infrared and color images of ancient paintings. Other methods are color weaving introduced by Hagh-Shenas et al. [15] or attribute blocks proposed by Miller [22]. Many approaches use color to indicate differences between 2D images. Since this is a very simple and intuitive way to display differences, it can be applied to various domains. Hollingsworth et al. [16] used a specific colorization scheme for difference images to compare 2D gas chromatographies. Sahasrabudhe et al. [31] used color coding to highlight differences between unequal visualizations, whereas Suomi and Oikarinen [34] concentrated on MRI datasets and Da Silva et al. [7] analyzed diffusion tensor volumes. Apart from color-coding, other methods of abstraction have been used to analyze image differences. Malik et al. [19] proposed a *multi-image view* for comparing images. Their technique subdivides the image space into hexagonal regions, and each region is subdivided into smaller elements which depict data from different series. This way contextual information is provided about the data, and outliers can be spotted very easily. The more elements a dataset consists of, the more subelements have to be created for every hexagonal region. At some point the sub-elements will be too small for a proper analysis, which makes this method unsuitable for large image datasets. The authors also state that their approach is targeted to grey-scale values only. Our approach for comparative image visualization is somewhat similar to the multi-image view, since it also aims at preserving information about the underlying data. However, due to the use of clustering, our method is scalable with respect to a significantly larger number of images. Our visualization technique is also only applied to regions where changes take place and therefore provides a better localization of differences.

Various approaches dealing with **parameter space analysis** also involve image comparison. Marks et al. [20] introduced *Design Galleries*, which allows the user to browse result images that have been produced by varying a given input-parameter vector. Ma [18] proposed *Image Graphs* to visually analyze the process of data visualization. *VisTrails* [32] is an interesting tool which provides visual comparisons for work-flows and images. Bruckner and Möller [2] developed a system to explore simulation parameters. They suggest to sample the given parameter space and then apply clustering to the output images to identify key changes. *Tuner*, a system developed by Torsney-Weir et al. [35], allows them to find proper parameter values for image segmentation by comparing the segmentation results to a ground truth. All these approaches rely on image comparison to evaluate output images.

Agglomerative **hierarchical clustering** is a statistical method of cluster analysis which aims at building a hierarchy of clusters [9]. In this bottom-up approach pairs of clusters are merged as one moves up the hierarchy. Hierarchical clustering has become a de facto standard for analyzing biological gene expression data in the past years [10]. It is also used in other domains, for example to analyze audio data as described by Clarkson and Pentland [4] or to classify ocean colors as proposed by Yacoub et al. [39]. We use hierarchical clustering to embed differences in the set of images in a hierarchy to identify different levels of data variances.

Focus+context and in-place interaction techniques give an overview over the available data, but also allow to further inspect details on demand. Several focus+context approaches in 2D and 3D can be found in the literature [5]. Mistelbauer et al. [24] proposed *Smart Super Views*, a tool for medical visualization which provides only the currently most relevant views to the user. Zhao et al. [40] implemented an interactive focus+context approach where the user can select regions of interest in 3D. Details about the ROIs are presented by preserving the context around the selection. *Contextual Snapshots*, presented by Mindek et al. [23], allow the user to keep track of spatial selections in visualizations. We use focus+context and in-place interaction techniques which enables the user to inspect individual image features.

3 Visual Analysis for Image Comparison

In this paper a new approach for comparative image visualization is presented. To effectively convey information about image differences, we decided to provide an interactive visualization system that preserves contextual information while enabling the inspection of individual *image differences*. According to Gleicher et al. [14], there are three common approaches used to compare data structures: side-by-side comparison (*juxtaposition*), blending (*superposition*) and explicit difference encoding (*aggregation*). VAICo uses a mixed approach of superposition and aggregation.

In our work we focus on comparison tasks which aim to identify and analyze differences in a large number (i.e., up to hundreds) of similar images. None of the images has to be defined as a reference image. Image differences are interpreted as variations in the image dataset (i.e., local color changes). In particular, VAICo is designed to assist users in identifying distinct classes of variations which characterize the underlying phenomena. Such investigations are common in medicine and biology, for instance. There they can provide important insights into the underlying cause of a disease, or the genetic mechanisms behind certain phenotypical variations.

3.1 Analysis of Image Set Differences

VAICo is based on an image comparison step where the image space, defined by the images in the given dataset, is divided into the two parts of *contextual information* and *regions of differences* (RoDs). Contextual information refers to parts of the image space that are the same in all images. RoDs correspond to variations in the image set. Image variations are interpreted as color changes between at least two images in the dataset. Image data related to a certain RoD is then embedded in a hierarchy, which enables the classification of changes in the data (Section 3.1.2). Figure 2 provides an overview of our approach. The results of the image comparison and clustering can be explored in our interactive visual analysis interface (Section 3.2) which is driven by the visualization of the RoDs.

3.1.1 Region-of-Difference Computation

In the first step RoDs have to be identified in image space. We propose two image comparison approaches which both result in a list of RoDs for the given image dataset.

To apply an **unbiased image comparison**, we employ a pixel-based image comparison metric based on the *Mean Squared Error* (MSE) as described by Zhou et al. [41]. All pixels from one image are compared to pixels at the same location in all other images in the dataset. A threshold is used to filter out low color variations. This enables users to control the algorithm's sensitivity with respect to changes in the data (see Figure 18 for further information). After applying the MSE calculations to all pixels in image space, a set of *difference pixels* is identified which represent pixels with varying color values in the images. Region growing [33] is then used to group difference pixels together to form disjoint subsets of pixels. Difference pixels are grouped together based on their spatial arrangement. In the region growing approach all difference pixels are considered to be seeds which potentially could start a new region. The region growing is initiated with a randomly selected pixel. Then neighboring pixels (according to an 8-connected neighborhood) are merged into a connected region. The process is iterated until all pixels have been assigned to a region, as demonstrated in Figure 3. The subsets resulting from the region growing step define the RoDs which form the basis of our interactive visual analysis approach. RoDs indicate the locations where changes take place in the data. All together, they give an overview on how much of the image space is affected by differences in the dataset.

In addition to the unbiased image comparison we also propose a **biased image comparison**. This enhances the image comparison by allowing us to include prior knowledge about the data. For some datasets not only image variations are of interest for the analysis, but also differences of *features* in the image. Features are represented as connected image regions which can have different characteristics in the image data (i.e., different color values). In contrast to the unbiased image comparison, RoDs are calculated for every image in the dataset *individually*. We employ a color segmentation approach based on Mean Shift [6] to identify regions of interest in every image in the dataset. The segmented regions then define a set of m RoDs per image in the dataset. Afterwards image comparison is applied, which in this case refers to sets of RoDs being compared to each other. The goal of the comparison is to find out whether RoDs of different images represent the same information. Figure 4 illustrates the process of RoD comparison. RoDs are compared based on their spatial position, shape and size. RoDs of different images are considered to represent the same region if their overlap exceeds a threshold (we use 90% as a default). The biased image comparison leads to a final list of RoDs for the dataset, where every RoD is present in at least one image. Using the biased instead of the unbiased image comparison allows the user to eliminate certain regions (e.g., background information) from further analysis.

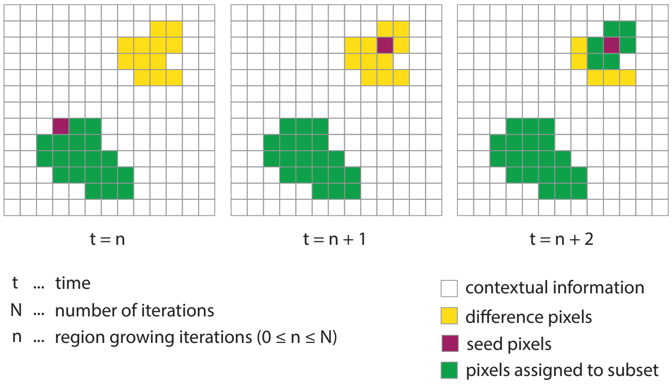
t = n
t = n + 1
t = n + 2
t ... time
N ... number of iterations
n ... region growing iterations (0 ≤ n ≤ N)
□ contextual information
🟨 difference pixels
🟪 seed pixels
🟩 pixels assigned to subset

Fig. 3. Illustration of subset computation by region growing. After the image comparison step a set of so-called *difference pixels* has been identified. Region growing is used to group them together into disjoint subsets. During region growing, difference pixels are assigned to the current subset until it cannot grow any more ($t = n$). Then another difference pixel is selected ($t = n+1$) and the iteration is continued ($t = n+2$).

The unbiased as well as the biased image comparison step result in a list of final RoDs that are used for further analysis. Since all images are registered and of the same size, a part covered by the RoD's location and size (defined by its corresponding set of difference pixels) can be found in every image. This leads to an unordered list of i image parts per RoD for i images in the dataset (Figure 5). In the case of biased image comparison, only images that contain a certain RoD are further considered. This leads to a list of l image parts per RoD in the case of biased image comparison, where $l \leq i$. These image parts are called *RoD-related image parts* in the following.

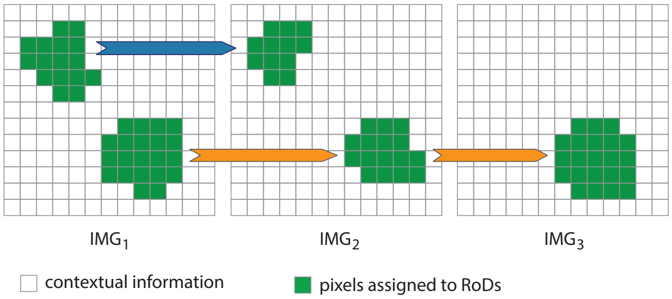

Fig. 4. Illustration of RoD comparison. After color segmentation a set of RoDs has been defined per image. In the comparison step RoDs of different images are identified that refer to the same information (i.e., overlap). In this figure the upper RoD can only be identified in the first two images (IMG_1 and IMG_2). The second RoD can be identified in all three images (IMG_1, IMG_2, and IMG_3).

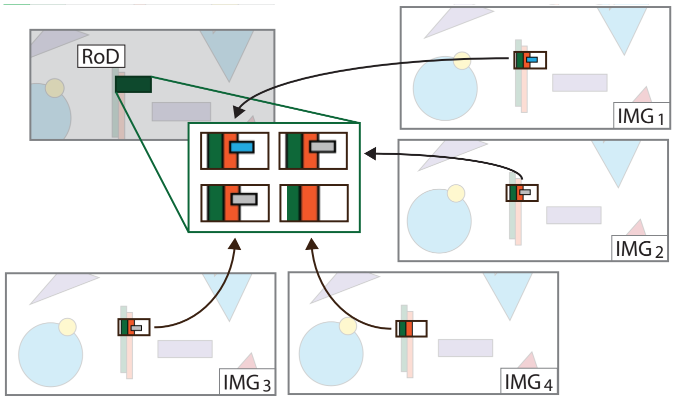

Fig. 5. Image data corresponding to a certain RoD (unbiased approach). In this Figure one particular RoD together with its corresponding image parts is displayed. The image parts are taken from all images in the set (as indicated by the arrows) and are defined by the RoD's size and location (i.e., the corresponding set of difference pixels).

3.1.2 Data Analysis by Clustering

In the second step we apply clustering to the RoD-related image parts to embed them in a hierarchy. This enables the classification of changes in the data and conveys information about the underlying data.

For the clustering, complete-linkage agglomerative hierarchical clustering [9] is used. Initially, every RoD-related image part forms a separate cluster. Then in each iterative step clusters are merged together according to their distance. The difference between two image parts is defined by their amount of pixel-wise differences (ranging from 0 to 100 percent). We employ a similarity metric based on the *Mean Squared Error* (MSE) as described in Section 3.1.1. If another image comparison metric should be used to compare the images, this metric should be employed here as well. The clustering terminates once all elements are included in one big cluster. Based on the clustering a tree is built which describes the results of the hierarchical clustering process (Figure 6).

Every hierarchy level in the hierarchical clustering tree describes a valid clustering result for the RoD-related image parts. However, for the best description of the given data, an *optimal* clustering has to be found. A clustering is basically defined by the number of available clusters. A clustering with maximum accuracy assigns each RoD-related image part to its own cluster. By comparison, a clustering with maximum compression uses one single cluster to include all RoD-related image parts.

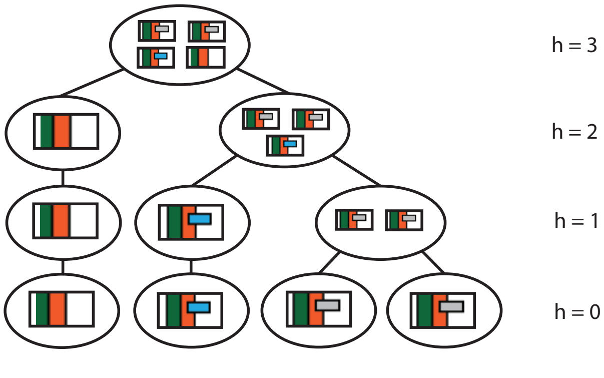

Fig. 6. Hierarchical clustering result for the RoD introduced in Figure 5. h depicts the hierarchy level in the tree. At the beginning ($h = 0$) all image parts are in separate clusters. In further steps ($h = 1, h = 2$) clusters are subsequently merged together according to their distance, until all elements are enclosed in one cluster ($h = 3$).

A clustering is considered to be optimal if it strikes a balance between the maximum accuracy and the maximum compression clustering [28]. An optimal clustering of the data will classify the image differences and provide more information about outliers as well as similarities in the data. To get an optimal clustering, the hierarchical clustering tree has to be cut at a certain level. We decided to use the *elbow criterion* as described by Pedersen and Kulkarni [28] to determine the tree level which contains the best clustering result. In essence, the elbow criterion specifies that a clustering should be chosen in a way that adding another cluster does not give a better modeling of the data.

In a clustering, outliers are defined by their distance to other clusters in the set. For our approach cluster outliers are of special interest, since they are considered to represent significant changes in the data. Therefore, clusters from the level with the best clustering result are sorted according to their inter-cluster distance. The cluster sorting process is done in an iterative way. In every step, the cluster with the maximum inter-cluster distance (defined by the sum of all distances to other clusters) is determined. It is then stored in an ordered list and excluded from further sorting operations. The process is repeated until no unsorted cluster is left. At the end, an ordered list of clusters is created for every RoD.

3.2 Interactive Visual Analysis Interface

The results of the image comparison step can be explored with VAICo's interactive visualization tools. Although the main interactions are based on the RoD visualization, it is possible to view the clustering and data analysis results as well. The main parts of the user interface are illustrated in Figure 7.

The interactive visualization elements are embedded in *image space* which is defined by the images in the given dataset. The entire set of images is visualized in one view. Parts of the image space which represent the same information in every image are depicted as **contextual information** (Figure 7a). We create an average image of all images in the dataset, and this image is displayed in the background. Pixels are faded out to enhance the visibility of the image variations. This contextual information is needed to embed the interactive visualization tools in the appropriate context.

RoDs are emphasized through **colored polygons** in the foreground (Figure 7a). The shape and position of the polygons correspond to the set of difference pixels assigned to the RoD (as described in Section 3.1.1). The RoD visualization provides a visual overview of the image comparison results. In combination with the contextual information, the RoD visualization allows the user to immediately differentiate between image variations and regions of similar information. In contrast to marking differences by abstract shapes like circles (Figure 1), the polygon shape provides more information about the extent of the image variations.

The number of RoD polygons in image space depicts how many image variations have been identified. The position and distribution of the RoD polygons shows where variations are located in the images, and how much of the image space is affected. In addition to size, shape and location of the image variations, information about the RoD-related clustering results is also included in the visualization. In case the clustering leads to a higher number of clusters than in other RoDs, the underlying RoD-related image data shows a greater variety than in other cases. This may indicate the existence of outliers in the data, wherefore these cases should be further analyzed by the user. Color-coding is used in the visualization to indicate a higher number of RoD-related clusters. A higher number of clusters is mapped to a darker color of the RoD polygon. Color-coding allows us to integrate the cluster information in the RoD visualization without occluding additional contextual information. In Figure 7a, the number of clusters for RoD_1 is lower than the number of clusters for RoD_2. Therefore, the color of RoD_2 is darker in the visualization.

RoD polygons can be inspected individually through mouse manipulation. When activated, RoD polygons are expanded to form **RoD widgets**. The widgets are displayed in image space in relation to the corresponding RoD polygons (Figure 7b). This embeds the widgets into the appropriate context and enables the user to analyze image

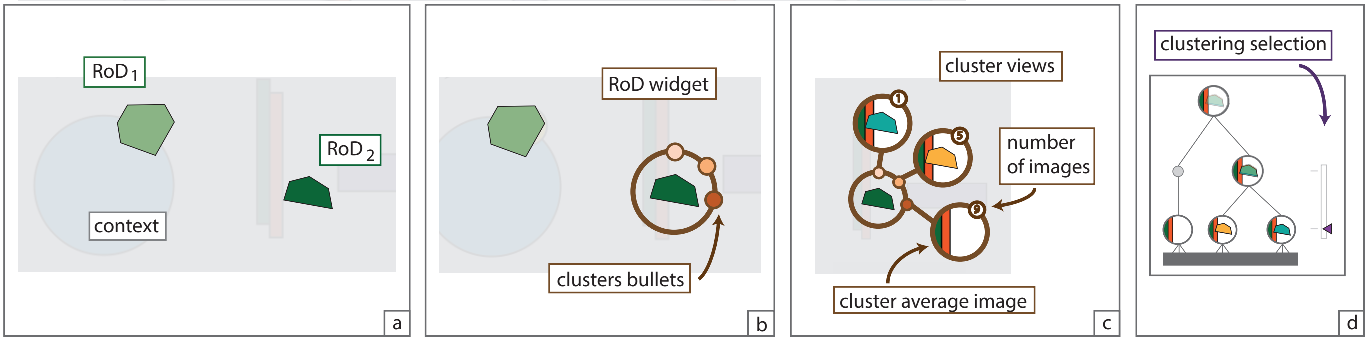

Fig. 7. Main components of the interactive user interface. The image space is displayed in one view (a). Similar information is displayed in the background (*context*) and the variations in the images are highlighted by RoDs. The RoDs can be further explored individually by using the assigned *RoD widgets* (b). The *cluster bullets* assigned to the RoD widget depict the clusters available for the corresponding RoD. The bullets can be expanded (c) to *cluster views* which consist of the assigned *cluster average images* and icons depicting the *number of images* in the cluster. This icon can be used to retrieve the list of images assigned to the cluster. In addition, the *clustering selection* (d) gives an overview over the hierarchical clustering tree and allows the user to select a new clustering (see also Figure 8).

variations without switching between different views. RoD widgets consist of a circle with colored bullets arranged around it. The colored bullets of the RoD widgets depict the corresponding clusters. The color of the **cluster bullets** changes according to the number of images that are included in the respective cluster. The color hue of the RoD polygon is always different from the hue of the bullets to prevent ambiguity. The color of a bullet is the darker, the more images the corresponding cluster contains. The bullets are ordered according to the number of images in the clusters. The user can decide whether they should be ordered in descending or ascending order. This allows the user to decide whether the cluster with the highest number of images should be at the top (i.e., to find patterns in the data), or whether the cluster with the lowest number of images should be first (i.e., to find outliers in the data).

The cluster bullets of the RoD widgets can be further expanded by mouse manipulation (Figure 7c) to display the **cluster views**. Clusters are then represented by the average image calculated from the images in the clusters. This immediately gives an overview on the image data that is encoded in the cluster. To get more quantitative information about the cluster size, the number of images is depicted for every cluster in a separate icon. These icons are attached to the average cluster images and provide additional functionality which can be controlled by mouse manipulation. When activated, the list of images encoded in the cluster can be viewed in a popup-window. This allows the user to select groups of images in the original data based on certain patterns.

In addition to the condensed view on the data, every RoD widget provides means to further analyze the results of the hierarchical clustering process. Mouse manipulation of the RoD polygons can also be used to view a visualization of the complete **clustering tree** (Figure 7d). We use a dendogram approach, since this gives an instant structural overview of the hierarchical clustering results. A dendrogram is a tree diagram which illustrates the arrangement of clusters produced by hierarchical clustering. Clusters which are created by merging are represented by the average image computed from their leaves. Visualizing the clustering helps to understand how the final RoD-related clusters have been generated. Furthermore, the number of hierarchy levels in the tree indicates the diversity of the RoD-related image parts.

The clustering which is visible in a RoD widget refers to a specific level in the clustering tree. The elbow criterion has been used to select the clustering. This decision may not be accurate in all cases. We therefore enable the user to overrule this decision by simply selecting another level in the **clustering selection**. The corresponding RoD widget is immediately updated to represent the new clustering (Figure 8). The top level of the clustering tree contains only one cluster which includes all available images. At the bottom level of the clustering tree, all clusters only contain one image.

Additional **control tools** are provided in VAICo which the user can employ to influence the visualization. The color of both the RoD polygons and the RoD widgets can be changed according to given color schemes. To skip variations that are not of interest for the data analysis, and to prevent the visualization from getting overly cluttered, the user can employ command tools to hide individual RoDs. According to the size and shape of the image variations, it may happen that RoD widgets will cover a large area of the image space when activated. Therefore, the RoD widgets can be shrunk. In the case of a biased image comparison (Section 3.1.1), the semantics of the color coding are slightly different, since information is collected on the number of images the RoDs are present in. The darker the RoD polygon, the higher the number of corresponding image parts. This way the colors of the polygons depict outliers in the data that are either available in many, or in just a fraction of the input images.

With its interactive visualization tools, VAICo provides an overview of the image variations and allows the user to further explore them. Differences in the data are encoded visually, but also functionality is provided to go back to the original input data if necessary.

4 Implementation

The raw data for the presented approach comprise an unsorted set of PNG images. The pre-processing, consisting of an image comparison step and a data clustering step, has been implemented in JAVA. The interactive visualization tools have been implemented as an interactive web application. The widgets are embedded in an HTML5 canvas and interactions are done in JavaScript.

The cost of the pre-processing depends on the number and size of the images in the set. The run-time depends on the number of identified RoDs, since for every RoD hierarchical clustering has to be applied.

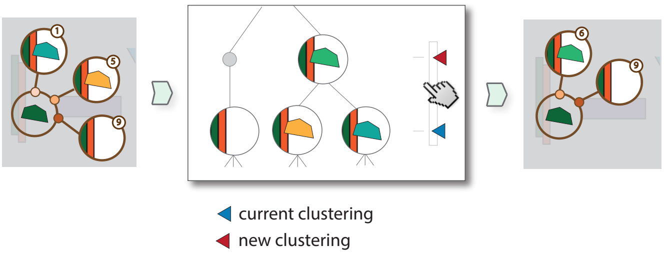

Fig. 8. Interactive clustering tree. In the interactive clustering tree the user can specify a new level by mouse manipulation. This will overrule the decision made during the clustering process. The new clustering is immediately available in the RoD widget.

The pre-processing for the largest data set called *Gene Expression*, which consists of 578 images of size 200x200 pixels (see also Section 5), takes 2.1 minutes. No user input is required during pre-processing. Afterwards the results of the pre-processing step are transferred to the visualization system by JSON files. The results are then loaded into the visualization system, which means that the provided JSON files are parsed. This takes 10 seconds for the largest data set. Afterwards the visualization system is ready to be used by the user. The interaction itself works in real-time. Our web-based architecture enables easy deployment of VAICo to explore and analyze online image collections and repositories.

5 Results

In order to evaluate the proposed comparative image visualization five different sets of images have been analyzed (Figure 9). The images are coming from different domains to show the applicability of the proposed method to different types of datasets.

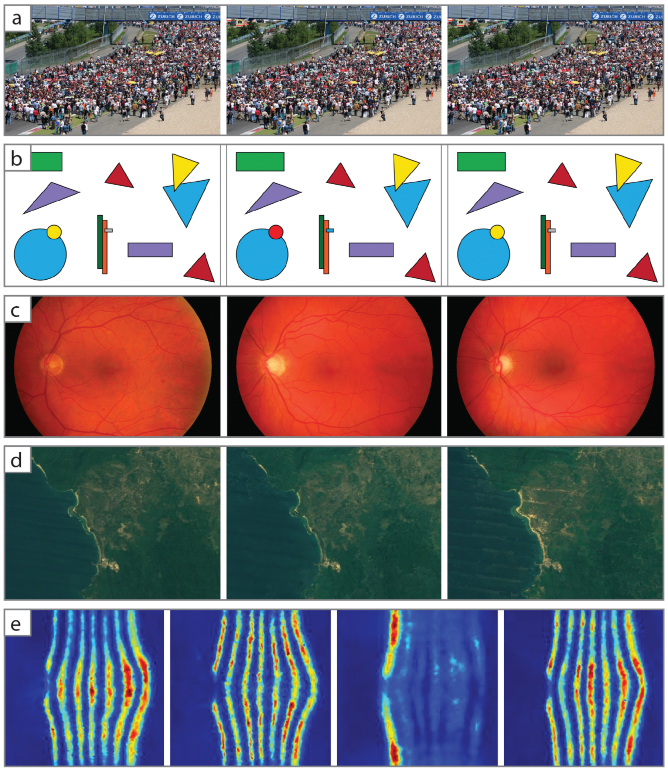

Fig. 9. Image datasets. The dataset *Puzzle* contains pictures of a real-world scene (a). The dataset *Shapes* contains images with shapes of different color. The dataset *Retina* contains retina images from different patients (c). The dataset *Satellite* consists of satellite images of a coastline in Indonesia (d). The dataset *Gene Expression* contains images with color coded gene expression information (e).

The selection of the first dataset called *Puzzle* was inspired by the well-known spot-the-difference puzzles that can be found in many newspapers. This dataset consists of 10 pictures which show a real-world scene. The images are of size 1050x700 pixels. In the images small elements disappear or change their color. The second dataset, called *Shapes*, is a synthetic dataset which consists of 52 images. The images are of size 600x400 pixels. They contain different types of shapes (rectangles, triangles and circles) of various colors and sizes on a white background. In the series of images, individual shapes disappear, re-appear or change their color. Analyzing the differences of both datasets by juxtaposition (i.e., placing images side-by-side) would be a tedious task for the user. This is primarily due to the number of images per dataset. Approaches that use juxtaposition do not scale well.

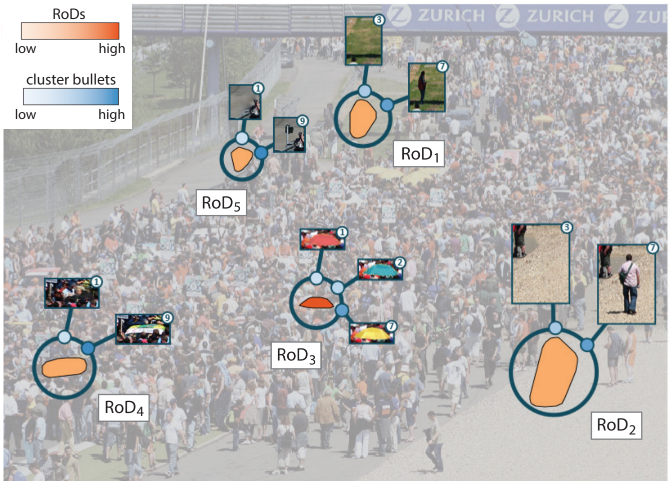

Fig. 10. Results for dataset *Puzzle*. Our approach identified five RoDs which depict the data changes in the images. One object in the scene changes its color (RoD_3), and some other objects are not present in all images (RoD_1, RoD_2, RoD_4, RoD_5). The color-coding of the RoDs shows the number of corresponding clusters. The color-coding of the cluster bullets indicates the number of assigned images.

We used the unbiased (i.e., pixel-based) image comparison approach to locate the image differences as described in Section 3.1.1. VAICo then takes the results of the comparison to give an instant overview of all variations in the datasets (Figure 10 and Figure 11). The analysis identified several image elements that are not present in all images, and also detected object color changes.

The third dataset is called *Retina* and contains 20 retina images of different patients. The images are of size 1168x779 pixels. In some of the images anomalies are present, because the patients suffered from diabetic retinopathy. Retinopathy is damage to the retina caused by complications from diabetes. Blood vessels at the back of the eye can bleed and blur vision. The leakage of blood can be seen as dark spots in the images. When analyzing this dataset, the unbiased image comparison approach was used as well.

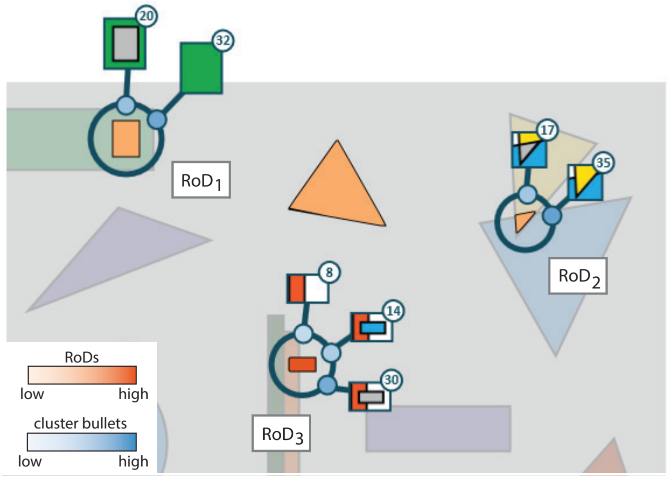

Fig. 11. Results for dataset *Shapes*. This dataset consists of images containing shapes of different color on a white background. Our approach identified five RoDs (three of them expanded in this Figure). Some of the objects are not present in all images (RoD_1,RoD_2), and some of them also change their color (RoD_3). The color-coding of the RoDs shows the number of corresponding clusters. The color-coding of the cluster bullets indicates the number of assigned images.

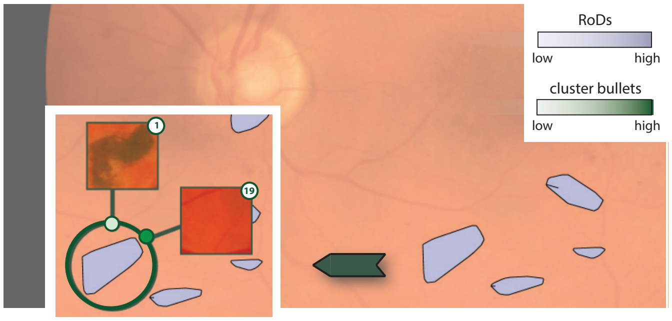

Fig. 12. Results for dataset *Retina*. This dataset consists of retina images from different patients. The image comparison identified dark spots on the retina, which can be further analyzed by using the RoD widgets. The color-coding of the RoDs shows the number of corresponding clusters. The color-coding of the cluster bullets indicates the number of assigned images.

In the comparative visualization of VAICo, dark spots on the retina can be identified as outliers in the data (Figure 12). Images without anomalies are included in one cluster since they contain similar information. The RoD widget allows us to explore the variations in the data, so that anomalies can be distinguished from normal data variations that do not indicate pathological changes. The RoD widgets allow us to retrieve the original image data, which keeps track of the investigated patients.

The set *Satellite* consists of 12 satellite images of size 614x450 pixels. They cover a time period from 2000 to 2011. One image has been produced every year. The images show a coast-line in Indonesia which has been affected by a tsunami in 2004. In these images, outliers like the damage caused by the tsunami which are present in only one image, can be easily missed. Additionally, differences between the state of the coast-line before and after the tsunami are not inherently available. Since no prior information was given about the structure of the image data, we used the unbiased image comparison approach. It was necessary to adjust the threshold to skip small changes in some parts of the data.

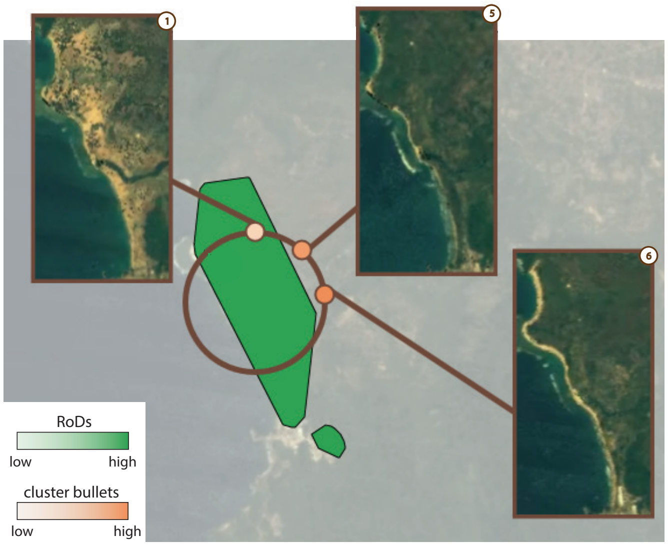

Fig. 13. Results for dataset *Satellite*. The images in this dataset show a coast-line in Indonesia. Damage caused by the tsunami in 2004 is identified as an outlier in the data (a). The images showing the state of the coast-line before (b) and afterwards (c) are summarized in separate clusters. The color-coding of the RoDs shows the number of corresponding clusters. The color-coding of the cluster bullets indicates the number of assigned images.

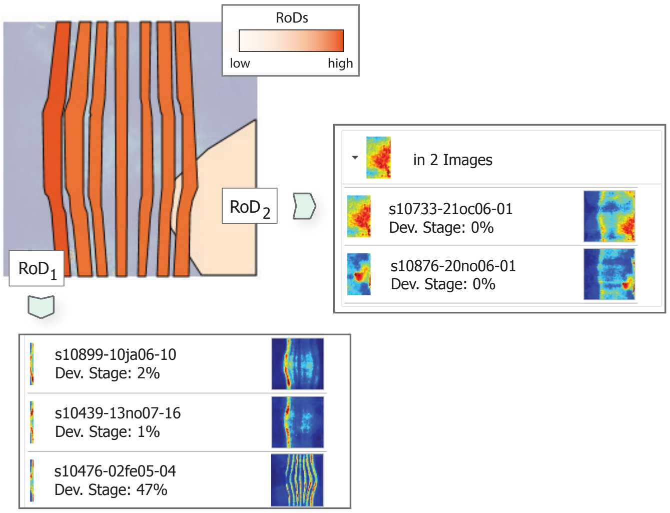

Fig. 14. Results for dataset *Gene Expression*. The color-coding of the RoD widgets depicts in how many images the corresponding RoD can be found. The first stripe (RoD_1) is available in more images than other stripes. Furthermore, an outlier region RoD_2 can identified. This region corresponds to unexpected gene expression data and only shows up in two images. The color-coding of the RoDs shows the number of corresponding clusters.

When viewing the comparison results with VAICo, the image part showing the coast-line is covered by one RoD. The damage caused by the tsunami on the coast-line is clearly visible as an outlier in the RoD data (i.e., separate cluster). In the remaining clusters the images before and after the tsunami are summarized (Figure 13).

The fifth set of images called *Gene Expression* is from the biological domain and consists of 578 images showing gene expression data. The images are of size 200x200 pixels. The data has been created from point cloud datasets of different fruit fly embryos as described by Fowlkes et al. [12]. In the images gene expressions of the *EVE* protein are color-coded from very high (red) to very low (blue). The *EVE* protein forms a pattern of seven vertical stripes of gene expression. Every image in the *Gene Expression* dataset corresponds to one fruit fly embryo. On the one hand, differences in the gene expression can be analyzed. On the other hand, outliers in the data can be detected which show unexpected gene expression patterns. However, due to the large number of images, a juxtapositional visualization does not scale well with this dataset. A superpositional comparison may hide differences in the gene expression of the seven regions.

We analyzed this dataset by using the biased image comparison approach as described in Section 3.1.1. We located the seven stripes of the *EVE* protein in the images by segmentation. Then the RoD data of the images was compared and the results were visualized with VAICo (Figure 14). The color coding of the RoD polygons indicates the number of images the RoDs are present in. In the visualization it can be seen that the first stripe shows up in more images than the other six stripes. This is due to the fact that some images in the earlier stage of the embryo development only contain one stripe. Additionally, outliers can be detected very easily in VAICo. They refer to patterns in the images where the gene expression does not look like as expected.

6 Evaluation

We collected user feedback to evaluate the presented visualization techniques. We hypothesized that using our visualization technique, participants would: (1) get results faster when searching for differences in a set of images; (2) get a better overview of individual variations in the image data; and (3) be able to better spatially localize differences in the image data. The results of the evaluation can be found in Figure 15.

For the feedback we compared a juxtapositional comparative visualization of an image set to the comparative visualization presented in

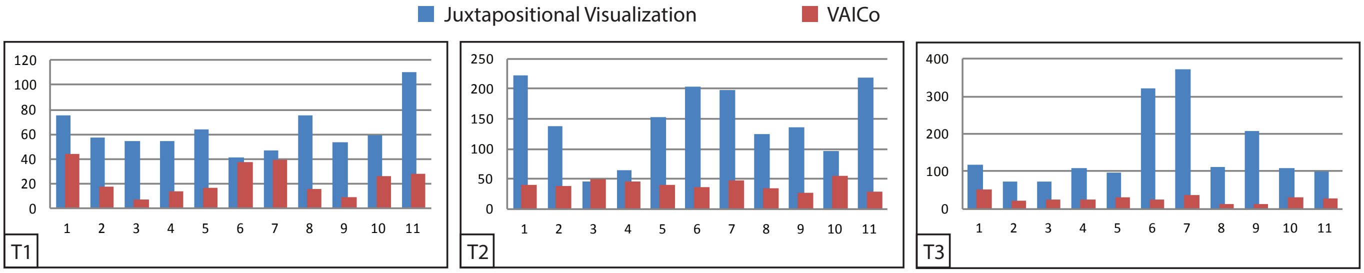

Fig. 15. Evaluation Results. The charts indicate the time it took the participants to complete the tasks T1-3. The x-axis depicts the participant number (1-11) and the y-axis gives the time (in seconds) it took to complete the task. Every participant had to complete every task twice: once by using a juxtapositional visualization and once by using VAICo. Therefore, two different measurements are available per task for every participant. The evaluation results show that VAICo clearly is of benefit for completing the examined tasks.

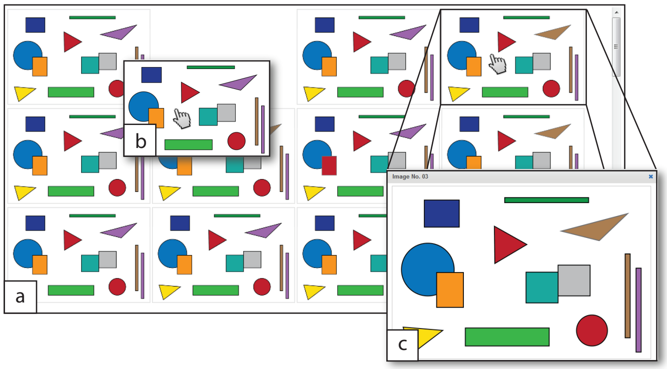

Fig. 16. Juxtapositional comparative visualization. In this visualization images that should be compared are placed side-by-side (a). Users can scroll through the list of images, sort images by drag-and-drop (b), and enlarge individual images by clicking on them (c).

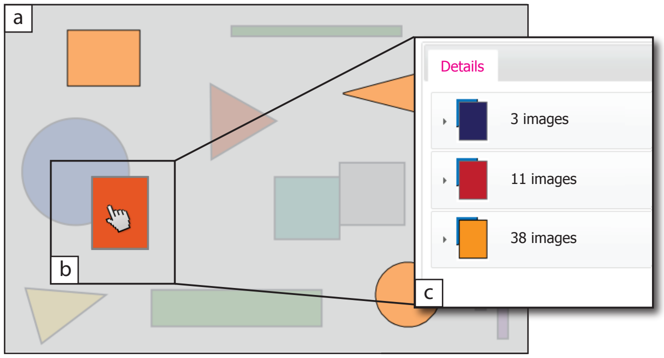

Fig. 17. Comparative visualization of images as employed for the user feedback. The set of images is presented in one view (a). Users can employ the RoD widgets (b) to get more details about individual image differences (c).

this paper. In the juxtapositional visualization participants could scroll through the list of images, sort them by drag-and-drop and click on images to enlarge them (Figure 16). VAICo, on the other hand, contained all the main features as described in Section 3.2 (Figure 17).

The images taken from the *Shapes* dataset contained shapes like triangles, circles and rectangles on a white background. Some of the shapes changed their color or disappeared in the image set. For the evaluation, six different dataset versions have been prepared. Due to their very general composition, images could be easily interpreted by users from different domains.

We designed three tasks which refer to the three hypotheses mentioned above:

- **T1: Depict variations with certain parameters**. In this task participants had to identify one shape which is present in all images. None of the remaining shapes did show up in all of the images.

- **T2: Analyzing local variations**. In this task participants had to identify one shape that is colored in three different ways in the images. Then they had to sort the three colors for this shape by the number of occurrences.

- **T3: Localization of changes**. In this task participants had to identify the most mutable image region (i.e., the image space part with the highest number of variations).

Participants had to complete all three tasks twice: once by using the juxtapositional comparative view and once by using the comparative visualization technique proposed in this paper. Datasets were switched between different tasks, and also between different visualization techniques. Therefore, participants never worked on the same dataset twice. The order of the tasks (as stated above) was the same for every participant. The order in which the two different visualization techniques were presented to the user was selected at random. For every task we measured the completion times and the correctness of the answers.

The user study was conducted with 11 participants from three different domains (computer science (7), engineering (3) and medicine (1)). Participants were between 26 and 57 years old. The group of participants consisted of 9 men and 2 women.

The results of the user study are presented in Figure 15. The results show that VAICo definitely improves the search time for image differences. In addition to time, the error rate was measured for every task. For the juxtapositional comparative visualization some participants made errors when trying to solve task T2 (error rate of 9.1 %) and task T3 (error rate of 27.3%). With VAICo participants did not give wrong answers. When analyzing the results, it can be seen that VAICo helps users to quickly get an overview on differences in the image set (task T1). It also helps to analyze local variations (task T2). Both tasks would require time-consuming analysis steps when done manually by the user. The comparative visualization brings significant improvements for users to localize variations in image space (task T3). This is a very tedious task if done manually on an unordered set of images.

7 Discussion and Limitations

VAICo operates on a large number of images and analyzes similarities and differences among them. As demonstrated in Section 6, image comparison tasks can be facilitated and accelerated by making use of our approach. VAICo is scalable to a large number of images and works with images from different domains (as outlined in Section 5).

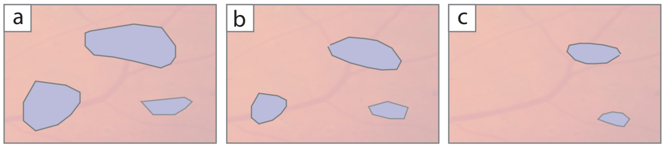

Fig. 18. Effect of sensitivity threshold. Three different threshold values increasing from (a) to (c) have been applied. Higher threshold values result in less image space occupied by RoDs.

Our approach nonetheless has some limitations, which point to interesting directions for future work.

VAICo is scalable to a large number of images. We use clustering to cope with scale and to identify patterns in the data. Clusters are depicted as bullets around the RoD widgets (as shown in Figure 7). If many clusters have to be displayed in the widget, there might not be enough space around the widget to visualize all of them. In this case additional functionality is provided to the user, which means that clusters are presented in a separate popup-window (where scrolling can be activated easily). However, we would like to point out that a high number of clusters bullets for a RoD (see Figure 7b) usually indicate that the underlying clustering should be refined. VAICo provides the possibility to adjust the clustering by selecting another level in the clustering tree (as described in Figure 6). We think that having the ability to refine the clustering is more appropriate for analyzing the data, than providing means to deal with high numbers of clusters. Including additional visualization techniques which can handle these special cases might be an interesting idea for future work.

The used *Mean Squared Error* (MSE) computation is a very simple image comparison approach which has some limitations. As one possibility to deal with these limitations, we provide the possibility to adjust the sensitivity of the algorithm. The user can define a *sensitivity threshold* for intensity changes. Figure 18 illustrates how the threshold effects the RoD calculation. The higher the threshold, the less pixels will be considered as variations in the data. Increasing the threshold will shrink the resulting RoDs, and might even cause RoDs to disappear. This way, particular intensity variations in the data can be skipped if they are not of interest for the data analysis.

MSE is very sensitive to global intensity shifts, and adjusting the sensitivity threshold might not be enough to deal with global changes in the data. Figure 19a shows an example of an image dataset where one of the images is lighter than the other ones. When applying MSE to this dataset, one big RoD is returned which covers almost the whole image space (Figure 19b). We used *Normalized MSE* [41] instead of MSE to compare the images. This solves the problem and returns the expected result (Figure 19c). In addition, this example shows that VAICo can be adapted to work with other image comparison metrics as well.

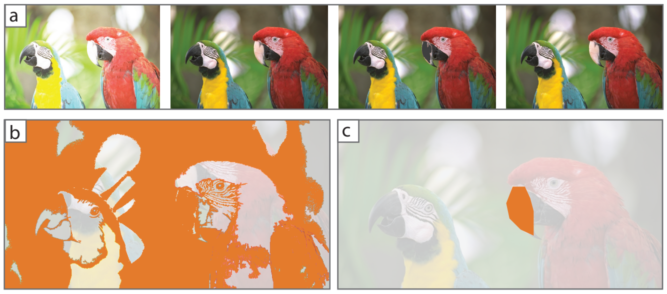

Fig. 19. Different image comparison metrics. One of the images in the given dataset shows more intensity variations than others (a). Since MSE is very sensitive to global intensity shifts (b), the approach fails in this case (differences are indicated in orange). To solve this problem and retrieve all desired RoDs, another image comparison metric (i.e., *Normalized MSE*) can be used (c).

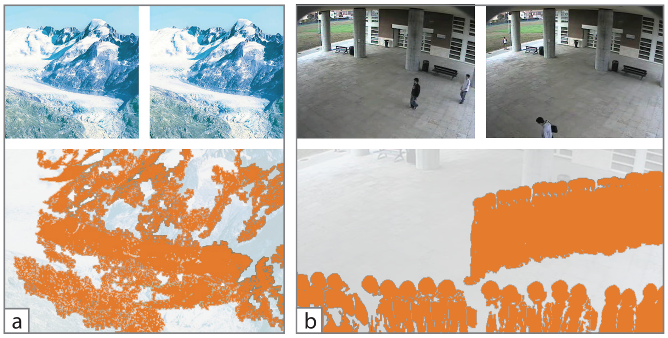

Fig. 20. Data analysis limitations. Landscape images (a), which show a high number or variations, and motion data (b) result in very large RoDs. This is because VAICo was designed for images that exhibit small local differences. Adjusting VAICo for other types of images is an interesting topic for future work.

VAICo operates on a set of images, where the individual images exhibit small local differences. The approach has its limitations if dealing with a high number of variations in the data. Figure 20 shows two examples where VAICo does not produce appropriate results, namely varying data in landscape images and motion data. VAICo is designed for scenarios where the obtained images have undergone a certain form of standardization. This is often done during the acquisition process itself (e.g., by employing a particular protocol for mounting and imaging) or through post-processing such as registration. As such procedures are highly dependent on the application domain, they are not the focus of our work. Instead, we assume that all images in our input set have undergone such a process, i.e., they are of the same size and represent similar regions in space.

In the case of landscape data (Figure 20a), an additional clustering step could be applied to the images, and VAICo could be used to further explore images in the individual clusters. In the case of motion data (Figure 20b), VAICo gives a quick overview on the motion tracks in the image. Although our approach is currently not suited to deal with time-dependent data, it will be interesting to explore its applicability for event detection in video data.

8 Conclusion

In this paper a visualization technique for the comparative visualization of multiple images has been presented. Interactive visualization tools are provided to explore the image space and drill-down on individual variances. Our visualization approach addresses the scalability of image comparisons and proposes ways to integrate contextual information and more detailed information in one view. Contextual information is preserved, whereas image variances can be efficiently spotted and put into context. Our approach can be applied to quickly identify small local differences in an image set. It is also helpful for analyzing the occurrence and value ranges of previously defined image features. We have demonstrated the scalability of our technique by applying it to five sets with varying numbers of images.

In the future, it will be interesting to explore how the proposed visualization techniques can be extended to image sets with spatially larger variations and what additional information can be extracted from the hierarchical clustering process. We also plan to investigate the applicability of VAICo for parameter space analysis. Another direction for further research is to extend our approach for the detection of key events in video data. Furthermore, we see the possibility to include our method in image tools like revision control [3].

Acknowledgments

The work presented in this paper has been partially supported by the ViMaL project (FWF - Austrian Research Fund, no. P21695) and by the AKTION project (Aktion OE/CZ grant number 64p11).

References

[1] E. Baudrier and A. Riffaud. A method for image local-difference visualization. In *Proceedings of the 9th International Conference on Document Analysis and Recognition*, volume 2 of *ICDAR '07*, pages 949–953, Curitiba, Paran, Brazil, Sept 23-26 2007. IEEE Computer Society.

[2] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, Oct 2010.

[3] H. Chen, L. Wei, and C. Chang. Nonlinear revision control for images. *ACM Trans. Graph.*, 30(4):105:1–105:10, July 2011.

[4] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '99, pages 3037–3040, Phoenix, Arizona, USA, Mar 15-19 1999. IEEE Computer Society.

[5] 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 2009.

[6] 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.

[7] 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.

[8] S. M. Drucker, G. Petschnigg, and M. Agrawala. Comparing and managing multiple versions of slide presentations. In *Proceedings of the 19th Annual ACM symposium on User Interface Software and Technology*, UIST '06, pages 47–56, Montreux, Switzerland, Oct 15-18 2006. ACM.

[9] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1973.

[10] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, Dec 1998.

[11] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, M. C. F. Oliveira, J. E. S. B. Neto, and R. Minghim. Multidimensional visualization to support analysis of image collections. In *Proceedings of the XXI Brazilian Symposium on Computer Graphics and Image Processing*, SIBGRAPI '08, pages 289–296, Washington, DC, USA, 2008. IEEE Computer Society.

[12] C. Fowlkes, C. L. Hendriks, S. Keränen, G. Weber, O. Rübel, M. Huang, S. Chatoor, L. Simirenko, A. DePace, C. Henriquez, A. Beaton, R. Weiszmann, S. Celniker, B. Hamann, D. Knowles, M. Biggin, M. Eisen, and J. Malik. A quantitative spatiotemporal atlas of gene expression in the drosohpila blastoderm. *Cell*, 133:364–374, 2008.

[13] D. Gareth and C. Min. Visualising video sequences using direct volume rendering. In *Proceedings of Vision, Video and Graphics*, VVG '03, pages 103–110, University of Bath, UK, Jul 10-11 2003.

[14] 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.

[15] H. Hagh-Shenas, V. Interrante, C. Healey, and S. Kim. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. In *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, APGV '06, pages 164–164, Boston, Massachusetts, USA, July 28-29 2006. ACM.

[16] B. Hollingsworth, S. Reichenbach, Q. Tao, and A. Visvanathan. Comparative visualization for comprehensive two-dimensional gas chromatography. *J Chromatogr A*, 1105(1-2):51–8, 2006.

[17] P. Kammerer, A. Hanbury, and E. Zolda. A visualisation tool for comparing paintings and their underdrawings. In *Proceedings of the Conference on Electronic Imaging for the Visual & Performing Arts*, EVA '04, pages 148–153, Florence, Italy, July 26-31 2004.

[18] K. Ma. Image graphs: a novel approach to visual data exploration. In *Proceedings of the Conference on Visualization*, VIS '99, pages 81–88, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[19] 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, Sept 2010.

[20] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 389–400, New York, NY, USA, 1997. ACM.

[21] 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.

[22] J. R. Miller. Attribute blocks: Visualizing multiple continuously defined attributes. *IEEE Comput. Graph. Appl.*, 27:57–69, May 2007.

[23] P. Mindek, S. Bruckner, and M. E. Gröller. Contextual snapshots: Enriched visualization with interactive spatial annotations. In *Proceedings of the 29th Spring Conference on Computer Graphics*, SCCG '13, 2013.

[24] G. Mistelbauer, H. Bouzari, R. Schernthaner, I. Baclija, A. Köchl, S. Bruckner, M. Srámek, and M. E. Gröller. Smart super views - a knowledge-assisted interface for medical visualization. In *proceedings of the IEEE Conference on Visual Analytics Science and Technology*, VAST '12, pages 163–172. IEEE Computer Society, 2012.

[25] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. In *Proceedings of the ACM SIGGRAPH*, SIGGRAPH '03, pages 453–462, San Diego, California, July 29-31 2003. ACM.

[26] H. Pagendarm and F. Post. Comparative visualization: approaches and examples. In *Visualization in scientific computing (H. Göbel and H. Müller and B. Urban)*, pages 95–108. Springer-Verlag, 1995.

[27] A. Pang and A. Freeman. Methods for comparing 3d surface attributes. *Visual Data Exploration and Analysis III*, 2656(1):58–64, 1996.

[28] T. Pedersen and A. Kulkarni. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL-Demonstrations '06, pages 276–279, New York, NY, USA, 2006. Association for Computational Linguistics.

[29] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. Comparative visual analysis of 2d function ensembles. *Computer Graphics Forum*, 31:1195–1204, Jun 2012.

[30] J. B. Procter, J. Thompson, I. Letunic, C. Creevey, F. Jossinet, and G. J. Barton. Visualization of multiple alignments, phylogenies and gene family evolution. *Nature methods*, 7:16–25, Mar 2010.

[31] N. Sahasrabudhe, J. E. West, R. Machiraju, and M. Janus. Structured spatial domain image and data comparison metrics. In *Proceedings of the Conference on Visualization*, VIS '99, pages 97–104, San Francisco, California, USA, Oct 24-29 1999. IEEE Computer Society.

[32] C. Scheidegger, H. Vo, D. Koop, J. Freire, and C. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13:1560–1567, Nov 2007.

[33] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Chapman & Hall, 2 edition, 1998.

[34] K. Suomi and J. Oikarinen. Visualization of changes in magnetic resonance image data. In *Proceedings of the Winter School of Computer Graphics*, WSCG '00, Plzen, Czech Republic, Feb 7-10 2000.

[35] T. Torsney-Weir, A. Saad, T. Möller, H. Hege, B. Weber, and J. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics*, pages 1892–1901, 2011.

[36] N. Tory, S. Staub-French, D. Huang, Y. Chang, C. Swindells, and R. Pottinger. Comparative visualization of construction schedules. *Automation in Construction*, 29(0):68–82, 2013.

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

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

[39] M. Yacoub, F. Badran, and S. Thiria. A topological hierarchical clustering: Application to ocean color classification. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '01, pages 492–499, Vienna, Austria, Aug 21-25 2001. Springer-Verlag.

[40] X. Zhao, B. Li, L. Wang, and A. Kaufman. Texture-guided volumetric deformation and visualization using 3d moving least squares. *Vis. Comput.*, 28(2):193–204, Feb 2012.

[41] Y. 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.