

Supporting Activity Recognition by Visual Analytics

Martin Röhlig, Martin Luboschik, Frank Krüger, Thomas Kirste, Heidrun Schumann *
Markus Bögl, Bilal Alsallakh, Silvia Miksch †

ABSTRACT

Recognizing activities has become increasingly relevant in many application domains, such as security or ambient assisted living. To handle different scenarios, the underlying automated algorithms are configured using multiple input parameters. However, the influence and interplay of these parameters is often not clear, making exhaustive evaluations necessary. On this account, we propose a visual analytics approach to supporting users in understanding the complex relationships among parameters, recognized activities, and associated accuracies. First, representative parameter settings are determined. Then, the respective output is computed and statistically analyzed to assess parameters' influence in general. Finally, visualizing the parameter settings along with the activities provides overview and allows to investigate the computed results in detail. Coordinated interaction helps to explore dependencies, compare different settings, and examine individual activities. By integrating automated, visual, and interactive means users can select parameter values that meet desired quality criteria. We demonstrate the application of our solution in a use case with realistic complexity, involving a study of human protagonists in daily living with respect to hundreds of parameter settings.

Index Terms: I.3.6 [Computing Methodologies]: Computer Graphics—Methodology and Techniques; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Theory and methods

1 INTRODUCTION

Decomposing multivariate time series into sequences of segments and assigning labels to these segments is a common task. A prominent example is the recognition and analysis of activities of human protagonists for ambient assisted living. The goal is to reconstruct activities from sensor-generated time series to allow researchers to study behaviors in a conceptual way.

Typically, automated algorithms are applied to compute and label the segments. Labels correspond to specific activities and segments denote associated occurrences and time periods. Although these algorithms are reasonably fast, they are limited in accuracy, particularly when exceptional patterns occur. Moreover, the algorithmic output is controlled by parameters, whose influence on the segmentation is hard to predict. For judging the determined activities reasonably, it is important to understand the interplay between input parameters, labeled segments, and recognition accuracies.

The traditional evaluation of activity recognition algorithms is commonly performed by investigating the results with statistical analysis methods, e.g., analysis of variance (ANOVA), and statistical plots, e.g., confusion matrices or interaction plots. This is successful for rather simple recognition strategies, but it barely covers all aspects of more complex approaches that require sophisticated

parameter tuning. In our use case (Sect. 3), we apply such a complex approach to activity recognition based on Computational State Space Models (CSSMs) [12]. CSSMs require extensive configuration of parameters, which is time-consuming and error-prone.

Visual analytics provides first approaches that already support users in visually inspecting the labeled segments, identifying specific patterns, and adjusting these results interactively [1]. Open issues are how to support users in (i) exploring hundreds of different parameter settings, (ii) comparing their influence, and (iii) increasing the accuracy of the computed results.

With this paper, which builds upon ideas first presented as a poster at IEEE VIS'14 [18], we introduce a novel visual analytics approach to address these issues. Our contributions are:

Novel Visual-Interactive Design: We present an enhanced visual approach (Sect. 5) that provides a comprehensive overview of parameters, labeled segments, and accuracies in an interlinked fashion. Coordinated interaction enables users to explore complex relationships, to compare different configurations, and to inspect the various aspects of the data in detail.

Comparison of Statistical and Visual Analysis: Statistical and visual methods (Sect. 4 and 5) support the data analysis in different ways. Therefore, we compare our visual analytics approach with purely statistical analysis. We identify benefits and limitations (Sect. 6), and discuss how open issues can be tackled in future work (Sect. 7).

2 RELATED WORK

Our work is related to visualizing segmented and labeled time series, and to visual analysis of parameter dependencies in general.

Visualizing Segmented and Labeled Time Series Segmentation results are commonly visualized as sequences of colored stripes. Color encodes the stripe label, and size represents the corresponding time period. This representation has been used to show segmentation results along with the time series data to inspect industrial processes [1]. Multiple segmentation results can be visualized to support comparison [21]. Similarly, a *sequence index plot* can be used to analyze multiple state sequences [8], which is common in sequential data analysis [5]. However, such plots become difficult to interpret for larger numbers of sequences. Generally, labeled time series can be interpreted as categorical data over time, for which various visualization approaches exist [2, 22].

Temporal categorical data can be analyzed using similarity measures [23]. Such similarity metrics can also be useful for sorting and comparing parameter settings, labeled segments, and accuracies over time. Simple sorting methods are based on lexicographic, averaged Euclidean, and cluster-based techniques. A prominent class of cluster-based techniques are self organizing maps (SOM). SOMs can also be used to deal with categorical data [6].

Visual Analysis of Parameter Dependencies Segmentation and labeling algorithms are generally governed by a set of parameters. Visual analysis is a common means to study the parameters' influence [19]. So-called *global-to-local* approaches are directly related to our work. They provide overviews of several pre-computed parameterizations and associated outcomes and allow for a detailed inspection on demand. However, only a few existing techniques address parameter-dependent time series data. An example is the

*University of Rostock, E-mail: {martin.roehlig, martin.luboschik, frank.krueger2, thomas.kirste, heidrun.schumann}@uni-rostock.de

†Vienna University of Technology, E-mail: {boegl, alsallakh, miksch}@cvast.tuwien.ac.at

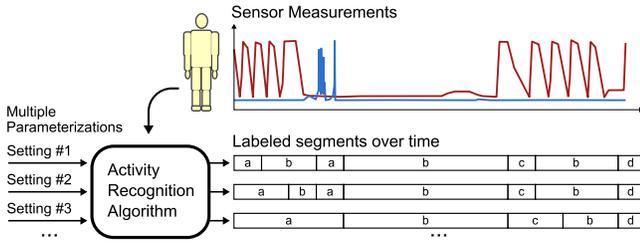


Figure 1: Illustration of the activity recognition process. The activities of a human protagonist are measured via sensors, then reconstructed by an automated algorithm for each applied parameter setting, and finally represented via different labeled segments over time.

approach for comparing parameter-dependent time series within larger parameter spaces [13]. However, the handling of categorical data has not been addressed so far.

Global-to-local techniques require the specification and computation of a large set of different parameterizations. Commonly, surrogate models are used to reduce calculation efforts [3, 20]. However, such models require that the data can be interpolated, which is not the case for our categorical time series. Other strategies for reducing parameterizations are for instance undirected optimizations [15] or parameter space partitioning [4], but these techniques do not provide a thorough overview of the entire parameter space.

Parameter-dependent data classification is also relevant in image segmentation. There it is used to link parameters with the 2D segmentation result [16] or with a ground-truth-based fitness function [20]. The latter approach also incorporates accuracy measures for judging the classification. Similarly, in [3] a ground truth is used to assess parameter-dependent multivariate predictions.

In conclusion, several individual techniques exist for visualizing either segmented and labeled time series or parameter dependencies. Yet, none of these techniques is directly suited for a combined analysis of parameters, labeled segments, and accuracies from activity recognition algorithms beyond statistical analysis methods. Therefore, we aim at developing an integrated approach that supports users in visually analyzing recognized activities over time, and that facilitates investigations of parameters' influence on the performance of the activity recognition. To achieve this goal, we (i) visualize multiple parameter settings along with labeled segments and associated accuracies, and (ii) provide coordinated interaction techniques to compare different aspects of the data and inspect details. Before we introduce our novel approach (Sect. 5), we first explain the examined activity recognition algorithm (Sect. 3) and the statistical analysis of its performance (Sect. 4).

3 ACTIVITY RECOGNITION AND USE CASE

Knowledge about current and possible future activities as well as eventually the final goals of individuals is the foundation for application domains such as security systems and assistive technologies. To this end, automated activity recognition algorithms are applied to assign activity labels to each time step of sensor observations. The basic idea is to abstract from the raw sensor data and to focus on detecting certain activities and reasoning about specific behaviors instead. However, to obtain adequate activity recognition results, the automated algorithms need to be configured and evaluated for different conditions and sensor systems. For judging the algorithms' performance in a given scenario, multiple algorithm runs with different parameter settings are necessary. Each algorithm run generates large data sets containing parameter-dependent information about the observed activities as segmented and labeled time series. Figure 1 illustrates this general activity recognition process.

As a concrete example, we consider an activity recognition algo-

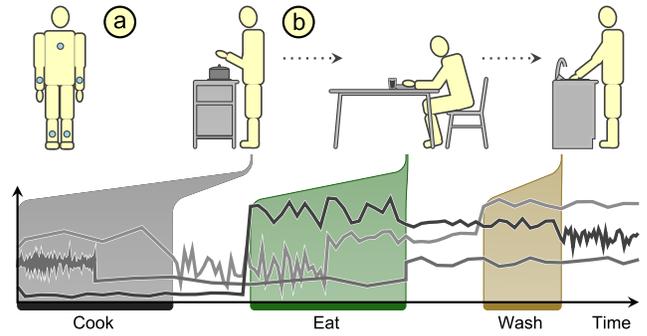


Figure 2: Illustration of our experiment. The human figure (a) shows, where the sensors are located. Three exemplary activities (b) – cook, eat, and wash – are depicted in combination with their time-oriented and multivariate sensor observations.

rithm based on Computational State Space Models (CSSMs) [12]. CSSMs are powerful tools for automatically segmenting and labeling time series data based on probabilistic graphical models. They can recognize activities even from noisy or ambiguous sensor measurements [12]. Although we specifically address activity recognition based on CSSMs, our approach is generic enough to be applicable to other algorithms and problems as well. An example showing the reconstruction of drilling processes is given in Sect. 6.

Studying and evaluating the performance of such sophisticated activity recognition algorithms is a difficult task. On the one hand, the algorithms' results are difficult to analyze and compare as they typically consist of long segmented and labeled time series with possibly multiple activities and thousands of time steps. On the other hand, the algorithms' results depend on the parameter configuration, where the impacts of individual parameters or their combinations are largely unknown. Our objective is to support machine learning experts in understanding the influence of parameters. The goal is to help them evaluate the algorithms' mode of operation and confirm or reject hypotheses about algorithm performance in realistic scenarios. Particularly, we aim at supporting experts in analyzing the parameter influence on the recognition performance (i) over time, (ii) of different protagonists, (iii) of all activities in general, and (iv) of individual activities.

The data for the evaluation of the examined CSSM was generated in an experiment with seven participants. Their activities included cooking a meal, setting the table, eating the meal, and finally cleaning the dishes (see Fig. 2). Each subject was instrumented with five inertial measurements units. Video logs were created and manually annotated to document the course of each experiment session and to obtain ground truth sequences for each subject.

A CSSM was created to recognize the subjects' activities by modeling the environment and possible actions [12]. As a result, we obtained 99 grounded actions (e.g., `take-knife-drawer`) from 16 activities (e.g., `take`) with a state space of about 146M states. Five parameters were defined to influence the CSSM:

- Sensor model (P_{Sensor}):** This parameter determines the quality of the sensor model and thereby the amount of information used during activity recognition.
- Filter mode (P_{Filter}):** This parameter selects different filtering modes for the processing of the sensor measurements.
- Goal distance mode (P_{Dist}):** This parameter offers different algorithms for a goal-directed selection of successive actions.
- Goal distance weight (P_{Weight}):** This parameter implements different degrees of goal-directedness when selecting actions.
- Duration model (P_{Time}):** This parameter offers different duration models for the activities to be recognized.

Single parameters	Subject 6 (all)		Subject 1 (<i>eat</i>)	
	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>
P_{Sensor}	1496.16	<0.001 ***	429.74	<0.001 ***
P_{Filter}	386.97	<0.001 ***	68.19	<0.001 ***
P_{Dist}	166.71	<0.001 ***	150.14	<0.001 ***
P_{Weight}	33.31	<0.001 ***	15.62	<0.001 ***
P_{Time}	9.96	0.003 **	6.34	0.014 *
Multiple parameters	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>
$P_{Filter} : P_{Sensor}$	32.25	<0.001 ***	26.07	<0.001 ***
$P_{Weight} : P_{Time}$	3.30	0.011 *	5.83	<0.001 ***
$P_{Sensor} : P_{Dist} : P_{Time}$	0.95	0.442	12.45	<0.001 ***
$P_{Sensor} : P_{Weight} : P_{Time}$	3.82	<0.001 ***	5.82	<0.001 ***
$P_{Filter} : P_{Sensor} : P_{Dist} : P_{Time}$	7.83	<0.001 ***	1.39	0.194
$P_{Filter} : P_{Dist} : P_{Weight} : P_{Time}$	1.05	0.426	1.92	0.016 *

Table 1: Exemplary results from statistically analyzing the performance of the CSSM. Shown are quantitative ratings (*F* and *p* values) and levels of significance (low: *, mid: **, high: ***) from applying a rANOVA to assess the influence of *single parameters* (top) and the influence of relationships between *multiple parameters* (bottom). The presented ratings are calculated regarding the general accuracy of the sixth subject and the accuracy of activity *eat* of the first subject.

To study the influence of the different parameters on activity recognition, we applied our CSSM to the sensor data. The domain experts identified 432 parameter settings to be examined. By applying the CSSM to the sensor data of all seven subjects we obtained 3,024 algorithm runs in total. For each run, the CSSM assigned one of the 16 activities to each time step in the sensor data. The overall duration of the experimental sessions varied per subject within the range of 160s to 320s (600 to 1,200 time steps at 3.75 Hz).

4 STATISTICAL ANALYSIS OF RECOGNIZED ACTIVITIES

To study activity recognition algorithms, statistical analysis commonly comprises: (1) deriving measures to assess recognition performance, (2) displaying these measures as statistical plots in relation to tested parameters for generating hypotheses about parameter dependencies, and (3) statistical hypotheses testing [12].

A typical measure for recognition performance is accuracy, that is, the percentage of correctly labeled time steps. The statistical analysis starts with calculating accuracy for each parameter configuration. Correct and incorrect labels are determined through comparing the sequences of recognized labels with the ground truth. To evaluate the influence of *single parameters* and of *multiple parameters* on the accuracy, we applied a repeated measures analysis of variance (rANOVA) across all tested configurations. We obtained accuracy ratings for single activities and globally for all activities.

Table 1 shows an excerpt of the statistical analysis. For *single parameters* (top), the results show that most parameters, except for parameter P_{Time} , have highly significant influence (marked with ***) regarding the overall recognition accuracy as well as regarding individual activities. Particularly, the sensor model (P_{Sensor}) has by far the highest influence, as indicated by its *F* value, followed by parameter P_{Filter} and P_{Dist} . Similar ratings of these parameters can be observed for other subjects and activities in our experiment.

Relationships between *multiple parameters* (bottom) show significant influence as well. For example, the relationship between parameters P_{Filter} and P_{Sensor} exhibits strong influence. Yet, the influence also varies depending on the activities and the relationships between parameters. For instance, the relationship $P_{Sensor} : P_{Dist} : P_{Time}$ shows significant influence on the recognition accuracy of activity *eat*, but less influence on the general accuracy. Some higher-order relationships between parameters, such as $P_{Filter} : P_{Dist} : P_{Weight} : P_{Time}$, have no significant influence.

In summary, all *single parameters* as well as certain relationships between *multiple parameters* clearly influence the performance and thus need to be considered for activity recognition based on CSSMs. However, applying purely statistical analysis to study how these dependencies manifest would require additional investigations. For example, suitable measures have to be defined for judging parameter influence on each activity and its temporal behavior. Moreover, large amounts of statistical ratings and associated plots would need to be compared to evaluate the influence of each parameter setting individually.

5 VISUAL ANALYSIS OF RECOGNIZED ACTIVITIES

Visual analysis is another way to study the performance of activity recognition. Next, we present a novel visual analytics approach for this purpose. We start by summarizing the design requirements.

5.1 Design Requirements

Discussing with domain experts and analyzing the statistics-based approach, we derived requirements for the visualization design.

Visualizing Labeled Segments (DR_1): The first step towards analyzing activity recognition is to inspect the recognized activities over time for *one* specific parameter configuration. The existence or absence of activities, their duration, and temporal sequence can help to decide about accepting or rejecting configurations. Thus, visualizing the labeled segments is a basic requirement.

Relating Parameters to Labeled Segments (DR_2): To explore interrelationships between different parameters, *multiple* or even *all* parameter configurations need to be examined at once. By comparing the influence of parameter combinations, users can search for appropriate configurations with regard to a given task. Hence, the second requirement is visualizing all relevant parameter settings along with the algorithmic outcome.

Relating Accuracies to Labeled Segments (DR_3): The results of the activity recognition are commonly compared to ground truth data to judge which parameter settings lead to adequate results. Deviations from the ground truth lead to reduced accuracies. Evaluating accuracy globally and individually for selected activities helps to understand how well certain parameter settings match the properties of the sensor data. Therefore, visually associating accuracy with the labeled segments has to be supported.

Exploring Patterns and Relationships (DR_4): A visual analysis of parameters, labeled segments, and accuracies, requires exploring global and local temporal patterns. *Global patterns* typically appear as similar outputs across all configurations or along an entire time series. Relating global patterns to the parameter settings helps, for example, to exclude individual parameters with less influence. *Local patterns* on the other hand, depend on specific ranges of parameter values, exist for limited time spans, or occur only with certain sequences of activities. Relating local patterns to the parameter settings helps, for instance, to detect parameter values with undesired side effects. Identifying global and local patterns is vital for choosing suitable parameter settings as well as for reasoning about the performance of the activity recognition algorithms. Hence, besides visualizing the data, interactive exploration of patterns and relationships needs to be facilitated.

5.2 Visual Encoding

Our main objective is the exploration of the parameters' influence on labeled segments and associated accuracies over time. For this purpose, we adapt the visual layout found in [13]. We extend this design to (i) consider categorical data, (ii) take accuracy information into account, (iii) show aggregated information about segmentation results both per time step and per algorithm run, and (iv) integrate tailored interaction techniques to explore the data, parameter settings, and relationships. To this end, we provide a compact visual overview (see Fig. 3) to match the design requirements $DR_1 \dots DR_4$.

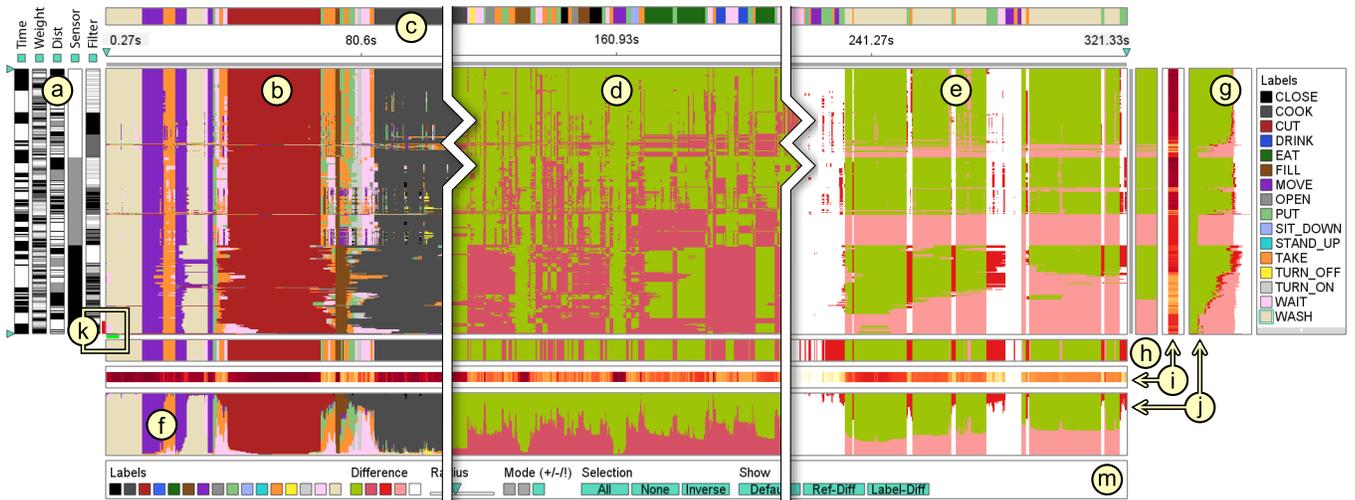


Figure 3: Schematic illustration of the visual analysis tool. For each parameter setting (a) three different modes are shown: activities as color-coded stripes (b), deviations from the ground truth (c) for all activities (d), and for one selected activity (e). Two additional views represent three types of aggregated information per column (f) or per row (g): the dominant activity (h), the percentage of the dominant activity (i), and the percentages of all activities as stacked bars (j). A visual cue (k) denotes the presence (red) or absence (green) of over-plotting in rows and columns. Selecting and highlighting different parts of the data is enabled through a dedicated toolbar (m).

Encoding Labels We select the visual variable *color* to encode the labels as it is well suited for representing categorical data [14] and allows for a compact visual design (cf. [11]). Other reasonable options are either needed to represent the temporal dimension, e.g., *position*, or would require more display space, e.g., *area* or *length*. To address scalability issues when dealing with larger numbers of labels, we select distinct colors from the $L^*a^*b^*$ color space and maximize the perceptual distance of neighboring labeled segments by permuting their color assignment (cf. [9]).

Visualizing Labeled Segments The algorithmic outcome is visualized as sequences of colored stripes over time (see Fig. 3b). As aforementioned, we apply qualitative color palettes with every label assigned to a distinct color. Additionally, the preset color coding can be interactively adapted to match given color conventions.

Each row of colored stripes represents the segmentation and labeling results of one algorithm run. This compact encoding provides a good overview and facilitates the comparison of multiple segmented and labeled time series (DR_1).

Visualizing Parameters The parameter settings are shown aligned to each algorithm run (Fig. 3a) and thus, the connection between parameter values and the algorithmic outcome is established (DR_2). Each row of the parameter display encodes one particular parameter setting and each column represents a single parameter. The parameter values are encoded using shades of gray. For each parameter individually we use distinct shades for categorical parameter values or a sequential scale for numerical parameter values. This way, the parameters can be clearly distinguished from the presentation of labeled segments.

Visualizing Accuracies To evaluate the activity recognition performance of each parameterization, we visualize accuracy information along with the labeled segments (DR_3). As a reference, the ground truth data is visualized as a single sequence of colored stripes on top of the labeled segments (Fig. 3c). To show deviations to this ground truth data, the user can switch between two additional encoding modes for the labeled segments. The first mode shows the accuracy of the labeled segments in a binary encoding: correct segmentations are colored green and incorrect are colored red (Fig. 3d). The second mode provides a more detailed differentiation regarding incorrect results of *one* selected activity: true

positives (TP) are colored green and false positives (FP) and false negatives (FN) are colored in different shades of red (Fig. 3e). This way, the color coding of both modes is visually consistent. Selecting another activity in the second mode results in a re-coloring of the labeled segments.

Compound Views of Labeled Segments The visual encoding discussed so far, allows for exploring individual segmentation and labeling results, associated accuracy information, and underlying parameter dependencies. To summarize the labeled segments in different ways, we introduce a horizontal and a vertical compound view. Both views provide a greater overview of multiple segmentation results at once and in this way facilitate the detection of local and global patterns. The horizontal compound view aggregates labels across all parameterizations for every point in time. It is located beneath the visualization of the labeled segments (Fig. 3f). The vertically oriented compound view is positioned on the right of the labeled segments and sums up labels for the entire time series of each parameter setting (Fig. 3g). Both compound views encompass three separate components. The first component encodes predominant labels by calculating majorities over parameter settings and time (Fig. 3h). The second component communicates the percentage of this labeling (Fig. 3i). The percentage values are obtained by computing the ratio of dominant labels with respect to the overall distribution of labels. These values are then encoded using a sequential color scale, with high values indicated by dark colors. To clearly identify high and low percentages, we enlarge respective stripes outside of a certain percentage interval. The third component provides further details and encodes the ratio of each label using stacked bars (Fig. 3j). Labels with very low ratio of segments are summarized into one neutral color to ensure visibility and to make the ratios comparable.

In this way, the three components of the compound views allow to easily trace predominant labels and accuracy conditions. For example, the horizontal compound view gives the user a sense for the stability of the label selection over time. Moreover, it facilitates an interactive data folding: labels, which are of high accuracy and stable across different parameterizations for long time periods, can be collapsed to focus the exploration on uncertain segments.

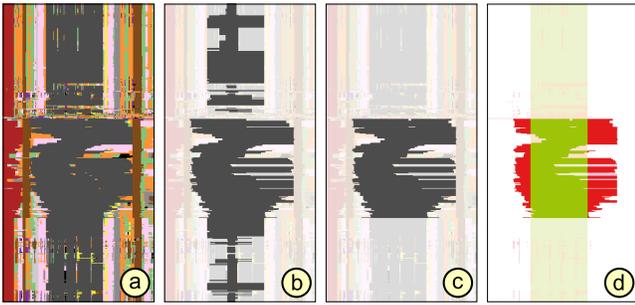


Figure 4: Selecting and highlighting labeled segments. The image sequence shows: (a) the default activity view, (b) a selection of segments for one activity (gray) crossing one time point, (c) a reduced selection for certain parameter settings, and (d) the same selection in accuracy encoding mode with true (green) and false (red) positives.

5.3 Interactive Exploration

To support a comprehensive visual exploration we provide a rich set of tailored interaction techniques. These techniques are fundamental for studying patterns and relationships in changing scenarios and thus, particularly address design requirement DR_4 .

Navigation Our compact visual encoding provides overview for comparatively large amounts of labeled segments from different parameter settings. Yet, considering the size of the data in certain usage scenarios, showing all parameter values and time steps for each algorithm run can easily exceed the available screen space. Interactive zooming and panning enable users to overcome these limitations and view different subsets in greater detail. The parameter axis (vertical) and the time axis (horizontal), can be scaled and translated independently or in combination. During zooming and panning, all views are interlinked to preserve the relationships between parameters, labeled segments, and summaries in the compound views. Furthermore, each view can be moved and collapsed individually to improve screen space utilization.

Zooming and panning supports studying global patterns. Local patterns generally benefit from combining overview and detail displays. An interactive rectangular fish-eye lens [17] shows selected subsets in detail while preserving their context regarding the whole data set. Lens position and magnification can be adjusted to focus on the information relevant for the task at hand.

Visual cues provide additional support. Scroll bars represent the location of the current view and two visual indicators inform the user whether the visualization is affected by over-plotting along either axis (Fig. 3k). A red indicator signals that the perception of patterns might be impaired by over-plotting. Clicking an indicator switches to a more adequate zoom level.

Selecting and Highlighting Investigating patterns regarding selected labels or specific sequences in a data set of hundreds of parameter settings with thousands of labeled segments can be a tedious task. To support users, we extend our overview visualization with flexible selection and highlighting techniques.

A selection of labeled segments can be performed in several ways, including selecting (i) segments of labels, (ii) accuracy conditions, (iii) parameterizations, (iv) segments crossing a specific point in time, and (v) particular regions of interest by applying an interactive selection lens. Selections can be expanded or reduced using binary set operations, or be restricted to selected labels only. While selecting, the visual encoding is updated to dim deselected segments. All selection options are provided in a toolbar (Fig. 3m).

Our selection techniques support both: (1) changing the selection to compare different parts of the data and (2) viewing the selection in different encoding modes to analyze the chosen subset in greater

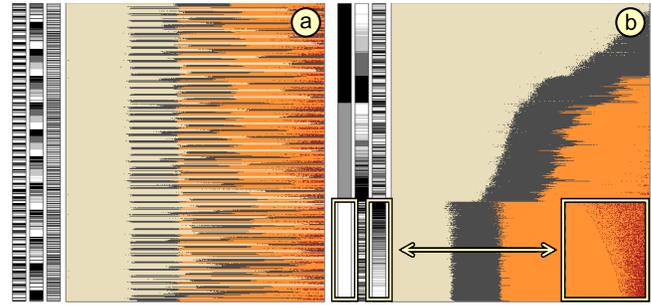


Figure 5: Sorting labeled segments. The overviews show algorithm runs (a) in their default order and (b) sorted based on labeled segments. The sorting reveals dependencies between the occurrence of red segments and values of the first and third parameter.

detail. Figure 4 illustrates this functionality. The default activity view is shown in Figure 4a. In the example, a selection is initialized by clicking on a bar within the stacked bar chart of the horizontal compound view to select all contributing segments crossing the specified time step (Fig. 4b). Afterwards, the selection is altered using the selection lens to include only segments of specific length (Fig. 4c). Finally, by switching the encoding mode, the selected segments are associated with accuracy information (Fig. 4d). This allows to examine true positively (green) and false positively (red) classified time steps. As we explain next, such selections can also assist in changing the arrangement of the visualization.

Sorting Visualizing parameter settings and labeled segments in a row-wise fashion enables analyzing their dependencies. However, the order of the rows is essential for discovering patterns across multiple configurations [7, 10]. To support the user in finding suitable orderings, we provide automated sorting based on (i) labeled segment, (ii) accuracy, and (iii) parameters.

Sorting based on labeled segments and accuracy helps to identify configurations with similar sequences, and to build hypotheses related to the associated parameter values. In turn, parameter sorting facilitates the interpretation of parameter influence, allowing us to test the hypotheses from the parameters' perspective.

Sorting multiple time series of labeled segments and parameter settings is rather challenging. Sorting labeled segments requires a similarity measure for sequences of categorical values that also considers their temporal context. Sorting parameter settings demands taking heterogeneous types of data into account, such as categorical and numerical parameter values with distinct scales and resolutions. To address such diverse data properties, we provide various metrics and algorithms for automated sorting. For example, we use modified edit distances in combination with self-organizing maps for sorting labeled segments over time. For parameters, we mainly employed lexicographic or averaged Euclidean based sorting algorithms. Fig. 5 illustrates the impact of sorting on the patterns revealed by our visualization.

In addition to global sorting, users can interactively restrict the sorting to selected parts of the data. Cursors highlight the selected parts in the overview and visually link them to parameter settings and the time axis. Optionally, different sorting methods can be applied to different selections. Also, rows can be reordered manually.

By means of automatic and interactive sorting, users can analyze value distributions and temporal patterns. For example, global and local dependencies between parameters and labeled segments can be identified by testing different sortings on subsets of the data and looking for patterns across rows. Sorting according to accuracy information allows to examine configurations with similar performance over time and to relate them to the ground truth by switching between activity and accuracy encoding modes. All selections and

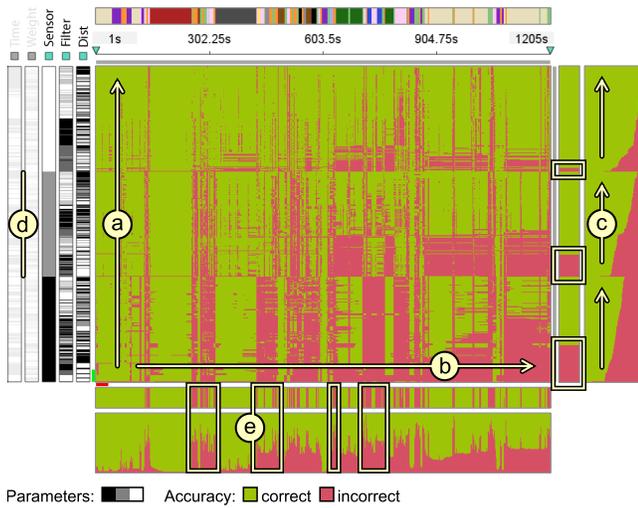


Figure 6: Visualizing correct and incorrect classifications. The overall accuracy mainly depends on (a) parameter P_{Sensor} , (c) parameter P_{Filter} , (d) parameter P_{Dist} , and (b) decreases over time. In addition, misclassifications accumulate (e) around certain activities in the ground truth (top). All effects can be confirmed by looking at the vertical and horizontal compound views, respectively.

sorting techniques are maintained in an *interaction history*. This allows users to directly undo or redo individual sorting steps and trace their overall analysis sessions.

Combining the visual design with the rich set of interaction techniques completes our visual analytics approach. Next, we apply this approach to analyze the data from our use case.

5.4 Results

We implemented a research prototype and applied it to analyze the data generated by the automated segmentation and labeling described in Sect. 3. To study the results, we conducted informal interviews with three experts from the visualization community. The experts had multiple years of experience in visual analysis of parameter spaces, multivariate time series, and probabilistic classifiers. Before each interview, we briefly explained the background of the data as well as goals and open questions regarding our use case. However, the results of the statistical analysis were concealed to prevent biasing interpretations of the shown data. After a short introduction of the research prototype, the visualization experts proceeded to explore and analyze the data. Assistance regarding the usage of specific features was provided and all comments and insights were documented by the interviewer. Our machine learning experts participated in the interviews as well. They were asked to reason directly about findings made by the visualization experts; some findings were reflected in subsequent discussions. Next, we summarize some of the findings and show respective visualizations.

Parameters' Influence on Accuracy Fig. 6 illustrates dependencies between parameters and the performance of the activity recognition algorithm in general. In the example, the data of the sixth subject is shown with labeled segments encoded according to correct (green) and incorrect (red) classifications. First, the time series were sorted in ascending order regarding the amount of correctly classified time steps. Based on the resulting overview, a subsequent sorting was applied with respect to the parameter with the most obvious influence. At first glance, Fig. 6 shows a global trend of low to high accuracy across algorithm runs from bottom to top (Fig. 6a) and a second global trend over time from left to right (Fig. 6b). The first trend (Fig. 6a) reflects the dependency of

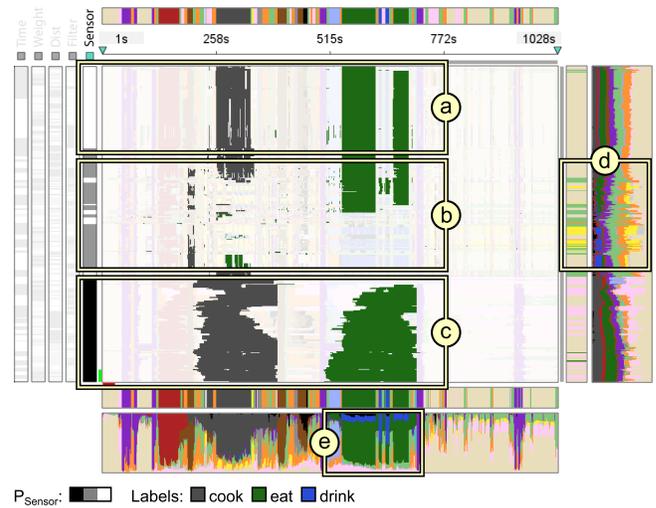


Figure 7: Visualizing activities `cook` and `eat`. The selection of all segments of both activities shows their dependence on the values of parameter P_{Sensor} : (a) segments are similar to the ground truth (top), (b) segments are missing for both activities, and (c) overlong segments. Moreover, the gray colored parameter value (b) leads to noisy segments with different labels (d).

the activity recognition results on the parameter P_{Sensor} . The global temporal trend (Fig. 6b) reveals that the number of misclassifications increases over time for the majority of parameter settings. In fact, this effect was suspected by our domain experts, but prior to visualizing the data it was not known for certain.

A second observation was made regarding the influence of P_{Filter} . Local gradients can be identified for P_{Filter} and the ratio of misclassifications per configuration (stacked bars in the vertical compound view (Fig. 6c)). This indicates that some filter modes mostly result in higher accuracy (light-gray and white) compared to other modes (black and dark-gray). Additionally, P_{Dist} shows an inverse local gradient for the labeled segments (Fig. 6d). Hence, higher accuracies within this band only appear in connection with two modes (black or gray), but not with the third mode (white).

Triggered by these observations, our domain experts raised questions concerning the influence of P_{Weight} on the accuracy. Hence, during one of the interviews a visualization expert sorted the visualization first with respect to P_{Weight} and then by overall accuracy. Interestingly, subtle patterns emerged while comparing the predominant labels in the vertical compound view with each band of configurations per parameter value. Increasingly good classification results coincide with lower parameter values, but accuracy decreases with higher values. This dependency is due to the fact that too low and too high degrees of goal-directed action selection can impair the activity recognition model. Consequently, finer sampling parameter P_{Weight} and testing respective parameter settings were noted for future improvements of the results.

Parameter Influence on Activities After analyzing the performance of the activity recognition in general, the analysis sessions proceeded to investigate parameter influence on individual labels. By relating accuracy to the ground truth, the visualization experts initially noticed accumulations of misclassifications around certain activities. An effect especially prevalent for the activities `cook` and `eat` across the data of different subjects (Fig. 6e). Hence, these activities were explored in greater detail.

As an example, Figure 7 depicts the data of the first subject with labeled segments encoded according to activities. All segments of `cook` and `eat` were selected and time series are sorted based

on their similarity. From the overview, three distinct groups of time series emerge that are vertically aligned with different values of P_{Sensor} . The first group (Fig. 7a) corresponds to the parameter value encoded in white and includes segments roughly equal to the ground truth. In contrast, numerous missing segments in the second group (Fig. 7b) indicate that the recognition of both activities fails with the gray colored parameter value. Finally, the black colored parameter value leads to the third group (Fig. 7c) of overlong segments which in case of activity `eat` even merge two separate instances in the ground truth. The vastly different results of these three groups hint at an overly coarse sampling of P_{Sensor} . Hence, studying other values might deepen the understanding of the parameter's influence. For example, testing other sensor models could help to obtain adequate recognition results for both activities.

The second group of missing segments (Fig. 7b) was investigated further. Selecting and highlighting the segments of other labels made clear that the respective parameter value had produced noisy segmentation and labeling results. To determine which labels were confused, the visualization experts examined the stacked bars in the horizontal compound view (Fig. 7d). Instead of `eat`, mainly the activity `drink` was recognized. Switching between accuracy encodings for each label and comparing false negatives of `eat` with false positives of the other labels confirmed these assumptions.

The results of the visual analysis suggest that parameter influence can be observed by global and local patterns in the computed activities and associated accuracy. Based on the visual analysis, the domain experts were able to answer open questions regarding dependencies between individual parameters values and the performance of the activity recognition. Several new hypotheses were generated on how the influence of single parameters and combinations of multiple parameters manifests. In addition, exceptional parameter settings and temporal influences were identified. All of the reported insights require further verification and validation, for example, by testing the parameter settings with different data sets.

6 DISCUSSION AND LESSONS LEARNED

So far, we described the statistical analysis approach for evaluating recognition performance (Sect. 4) and our novel visual analytics approach (Sect. 5) in isolation. In this section, we consider both approaches in a broader discussion.

Participatory Design Our visual-interactive design is the result of a participatory design process in cooperation with a group of four machine learning experts. In this process we conducted several collaborative analysis sessions. The gathered feedback formed the basis for our visualization and interaction techniques. Most design decisions were made to address data properties and analysis tasks. Some visualization components were motivated by the machine learning experts, e.g., the compound views for showing label distributions and the amount of correct and incorrect segments.

Reflecting on the Findings Generally, statistical analysis as well as visual-interactive exploration support users in understanding the complex interplay between parameters, labeled segments, and accuracies. However, the findings obtained are rather different. While statistical analysis provides quantified assessments of specific aspects, findings from visual-interactive exploration are rather of qualitative nature. For example, the results of the statistical analysis allow to rate and compare the influence of single parameters and of multiple parameters (Table 1). In addition, by testing every parameter configuration, certain accuracy characteristics can be traced precisely. But, this comes at the cost that *relevant* parameter settings have to be determined beforehand. In this regard, the analysis mainly supports the identification of particular dependencies and provides results with regard to predefined questions.

In contrast, visual-interactive exploration provides overview and details on demand, without the need to specify the *relevant* parameter settings in advance. This enables users to investigate global

and local patterns as well as their temporal context, which in turn helps to explain results of the statistical analysis. As an example, the statistical results indicate significant influence of P_{Sensor} on the accuracy (Table 1). This influence can be observed visually as well (Fig. 6). On top of that, the visual analysis reveals variations in accuracy due to differently labeled segments of specific activities for each value of P_{Sensor} (Fig. 7). Thus, interactively studying the data leads to new hypotheses and supports reasoning about additional influences on the activity recognition's mode of operation.

Yet, complex relationships can be hard to discern visually. For instance, during the interview sessions mostly relationships between two parameters and rarely up to three parameters have been identified. Moreover, the experts commented that some of the perceived patterns were rather vague and that they were uncertain about their findings at times. Consequently, they asked for computation means to verify their hypotheses. This suggests that statistical and visual-interactive analysis should be used in combination. This way, global and local patterns can be communicated, hypotheses about dependencies can be generated, and interesting findings can be validated in an intertwined fashion.

Reflecting on the Analytic Process We analyzed the outcome of activity recognition to determine dependencies between parameters (input) and labeled segments (output). However, our current analysis (statistical and visual-interactive) is mainly an off-line process after the recognition procedure. Hence, it can be considered outcome-oriented. Consequently, relevant inputs have to be either known in advance or have to be tested exhaustively.

Yet, choosing suitable parameter values to investigate their influence is challenging. Analyzing the inputs and outputs of activity recognition can support users in finding appropriate parameter samples. For instance, starting an evaluation by testing only subsets of parameter settings, allows to generate initial hypotheses. Domain experts can then reject parameters with insignificant influence in an early stage of the evaluation and instead focus on testing additional values of parameters which show strong effects. This enables decisions about the performance with regard to certain quality criteria, without the necessity of testing all parameter settings. This way, moving towards an online analysis of the data facilitates direct feedback and steering of the evaluation, which in turn helps in guiding users to desired results and fine-tuning algorithms to specific needs.

For example, the insights gained from the statistical and visual-interactive analysis in our use case led to further samplings of P_{Weight} to better understand its influence and to improve previous results (see Sect. 5.4). Our domain experts identified the ability to visually explore different patterns and to generate hypotheses early to be a major advantage. But, they simultaneously asked for tightly interrelating such visual analysis methods into the concept phase and statistical evaluation phase of their algorithms.

In this context, other open issues from applying automated activity recognition remain, such as dealing with diverging data properties, different algorithms, or changing performance requirements. For instance, our use case focused on studying parameters' influence for one algorithm (CSSM) with a fixed set of parameter settings in a given application scenario. The applied statistical analysis methods were customized to fit the evaluation of this specific setup. Consequently, they are hard to generalize regarding other conditions or evaluation constraints. In contrast, our visual analytics approach is generic enough to be applied to other algorithms or data sets for generating insights and hypotheses. Though, validating and verifying these hypotheses currently needs to be done separately.

Other Application Scenarios While our use case focused on data from activity recognition, the proposed solutions are applicable in other fields as well. Generally promising is the analysis of parameter influence on time series segmentation and labeling. For instance, Alsallakh et al. [1] present a scenario dealing with the reconstruction of drilling processes from sensor measurements using

automated methods. Similarly to our use case, their scenario involves long term time series containing thousands of time steps. In case of an erroneous segmentation and labeling of this data, drilling specialists have to inspect the reconstructed processes, adjust parameters of automated algorithms, and compare according outputs. In this regard, our solution can be beneficial as it provides overview for multiple parameter settings and allows for exploring different properties of the complex data. Moreover, by analyzing parameters' influence with our solution, the specialists can find suitable parameterizations and adjust the algorithms to different drilling conditions.

An interesting aspect of such applications is that the ground truth is often not known in advance. Concerning this matter, our approach enables comparing different segmentation and labeling results and choosing configurations that meet desired quality criteria. For example, using the overview and compound views predominant labels can be traced over time while also communicating the stability of this label selection. On this basis, users can adjust the results interactively to compensate for uncertain segments or missing data (cf. [1]). Once suitable labeled segments have been defined, they can be used as a reference, i.e., in place of a ground truth, to analyze other configurations in their context with our proposed solution.

7 CONCLUSION AND FUTURE WORK

In this paper we present a novel visual analytics approach for parameter-dependent activity recognition and compare our approach with statistical analysis methods. The statistical analysis of recognized activities enables rating the influence of parameters on the performance of used algorithms. Supported by our novel overview visualization of parameters, labeled segments, and accuracies, users can investigate the computed results and discover global and local patterns. Coordinated views and interactions help exploring and comparing different aspects of the data and facilitate hypothesis generation about associated dependencies. In conclusion, by providing extensive visual and interactive means, our approach supports users in understanding parameters' influence and in this way helps in increasing the accuracy of computed results.

To improve our approach, we plan further user studies and interviews with domain and visualization experts. Considering the lessons learned, we are interested in more tightly interrelating statistical and visual-interactive methods. For example, using direct visual feedback, users can generate hypotheses regarding analyzed parameter settings and trigger according statistical analysis to verify them. This allows to drill-down only into parameter subspaces that fit specific requirements, which reduces computational costs.

Additionally, our solution will be extended for comparing multiple data sets at once, such as recordings from several protagonists or different activity recognition algorithms. On the one hand, this allows to consider subjective and environmental influences when evaluating the performance of applied methods. On the other hand, taking the provenance of the data, i.e., changing algorithmic models, and their associated parameter spaces into account enables users to enhance their search to find an appropriate recognition algorithm with the best possible combination of parameters. Yet, this requires enhanced visual-interactive designs for dealing with the added complexity, e.g., dedicated aggregation methods to support scalability, but also for detecting and analyzing sophisticated relationships among several influencing factors.

ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) as part of VASSiB (SPP 1335), the DFG graduate research school 1424 MuSAMA, and by the Austrian Federal Ministry of Science, Research, and Economy via CVASt, a Laura Bassi Centre of Excellence (No. 822746).

REFERENCES

- [1] B. Alsallakh, M. Bögl, T. Gschwandtner, S. Miksch, B. Esmael, A. Arnaout, G. Thonhauser, and P. Zöllner. A visual analytics approach to segmenting and labeling multivariate time series data. In *Proc. of EuroVA'14*, pages 31–35, 2014.
- [2] K. Beard, H. Deese, and N. R. Pettigrew. A framework for visualization and exploration of events. *Inf. Visualization*, 7(2):133–151, 2008.
- [3] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Comp. Graph. Forum*, 30(3):911–920, 2011.
- [4] S. Bergner, M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad. Paraglide: Interactive parameter space partitioning for computer simulations. *IEEE TVCG*, 19(9):1499–1512, 2013.
- [5] C. Brzinsky-Fay, U. Kohler, and M. Luniak. Sequence analysis with stata. *Stata J.*, 6(4):435–460, 2006.
- [6] N. Chen and N. C. Marques. An extension of self-organizing maps to categorical data. In *Progr. in Art. Intelligence*, pages 304–313, 2005.
- [7] R. Fuchs, R. Peikert, F. Sadlo, B. Alsallakh, and E. Gröller. Delocalized unsteady vortex region detectors. In *Proc. of VMV'08*, pages 81–90, 2008.
- [8] A. Gabadinho, G. Ritschard, N. S. Müller, and M. Studer. Analyzing and visualizing state sequences in R with TraMineR. *J. of Statistical Software*, 40(4):1–37, 2011.
- [9] C. Hurter, M. Serrurier, R. Alonso, G. Tabart, and J.-L. Vinot. An automatic generation of schematic maps to display flight routes for air traffic controllers: Structure and color optimization. In *Proc. of AVI'10*, pages 233–240, 2010.
- [10] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo. LiveGantt: Interactively visualizing a large manufacturing schedule. *IEEE TVCG*, 20(12):2329–2338, 2014.
- [11] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE TVCG*, 6(1):59–78, 2000.
- [12] F. Krüger, M. Nyolt, K. Yordanova, A. Hein, and T. Kirste. Computational state space models for activity and intention recognition. a feasibility study. *PLoS ONE*, 9(11):e109381, 2014.
- [13] M. Luboschik, S. Rybacki, F. Haack, and H.-J. Schulz. Supporting the integrated visual analysis of input parameters and simulation trajectories. *Comp. & Graph.*, 39:37–47, 2014.
- [14] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM TOG*, 5(2):110–141, 1986.
- [15] J. Marks, B. Andalman, P. A. Beardsley, W. T. Freeman, S. F. F. Gibson, J. K. Hodgins, T. Kang, B. V. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. E. Seims, and S. M. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. of SIGGRAPH'97*, pages 389–400, 1997.
- [16] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle. Visualization of parameter space for image analysis. *IEEE TVCG*, 17(12):2402–2411, 2011.
- [17] U. Rauschenbach, S. Jeschke, and H. Schumann. General rectangular fisheye views for 2d graphics. *Comp. & Graph.*, 25(4):609–617, 2001.
- [18] M. Röhlig, M. Luboschik, H. Schumann, M. Bögl, B. Alsallakh, and S. Miksch. Analyzing parameter influence on time-series segmentation and labeling. Poster at IEEE VAST'14, 2014.
- [19] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE TVCG*, 20(12):2161–2170, 2014.
- [20] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE TVCG*, 17(10):1892–1901, 2011.
- [21] A. Vögele, B. Krüger, and R. Klein. Efficient unsupervised temporal segmentation of human motion. In *Proc. of SCA'14*, 2014.
- [22] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith. Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE TVCG*, 15(6):1049–1056, 2009.
- [23] K. Wongsuphasawat and B. Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *Proc. of VAST'09*, pages 27–34, 2009.