Making Parameter Dependencies of Time-Series Segmentation Visually Understandable

Christian Eichner, Heidrun Schumann and Christian Tominski
Institute for Visual and Analytic Computing, University of Rostock, Germany
{christian.eichner, heidrun.schumann, christian.tominski}@uni-rostock.de

Abstract
This work presents an approach to support the visual analysis of parameter dependencies of time-series segmentation. The goal is to help analysts understand which parameters have high influence and which segmentation properties are highly sensitive to parameter changes. Our approach first derives features from the segmentation output and then calculates correlations between the features and the parameters, more precisely, in parameter subranges to capture global and local dependencies. Dedicated overviews visualize the correlations to help users understand parameter impact and recognize distinct regions of influence in the parameter space. A detailed inspection of the segmentations is supported by means of visually emphasizing parameter ranges and segments participating in a dependency. This involves linking and highlighting, and also a special sorting mechanism that adjusts the visualization dynamically as users interactively explore individual dependencies. The approach is applied in the context of segmenting time series for activity recognition. Informal feedback from a domain expert suggests that our approach is a useful addition to the analyst’s toolbox for time-series segmentation.

Keywords: visualization, visual analytics, visualization

ACM CCS: • Human-centred computing → Visualization; Visual analytics; • Mathematics of computing → Time-series analysis

1. Introduction
Time-series segmentation transforms a time-varying signal into a sequence of labelled segments. This is a crucial data processing step in a variety of domains. For example, speech recognition requires audio streams to be subdivided into smaller chunks containing individual words. In medical applications, electrocardiography measurements must be segmented to find temporal phases with abnormal values within otherwise inconspicuous recordings. Activity recognition is another example where multiple sensor signals are segmented to extract time intervals and to assign to them the activity that an observed subject has carried out.

Algorithms for time-series segmentation usually provide parameters to tune the algorithm’s behaviour. The parameters can influence the segmentation in different ways. For example, a parameter could affect the length of segments or their position, moving them forwards or backwards in time. Also, the labels being assigned to the segments can depend on the applied parameter settings. However, it is often not clear upfront which parameter values lead to desirable segmentation results. Analysts need to develop an understanding of how the segmentation changes when certain parameters are adjusted and how re-parameterization might improve the segmentation.

Identifying parameter dependencies is challenging because the influence of parameters is not necessarily the same for all segments, labels and parameter values. There may be parameters that affect only particular segments or labels, while the majority of them remain unaffected. It could also be that only a particular range of parameter values shows an effect, while no effect can be observed from a global perspective.

In this paper, we describe a novel approach to support the visual analysis of parameter dependencies in the context of time-series segmentation. Preliminary results of this research have been published as a poster at the EuroVis conference [EST18]. Here, we contribute as follows:

• We introduce a feature-based description of segmented time series suitable for correlation calculations that aim to automatically determine dependencies between parameters and segments.
• We design overview visualizations of the calculated correlations that enable analysts to identify and compare all dependencies existing in the data.
• We provide additional support for automatically emphasizing relationships through highlighting and sorting of the visualization of the segmentation data, making them verifiable and understandable to analysts.

The key idea behind our work is to compute and visualize correlations between parameters and segmentations. Section 3 will introduce a set of features that characterize segmentations and serve as the basis for the correlation calculation. The actual calculation implements a subspace strategy to capture global as well as local dependencies.

With the help of the calculated correlations, the analysis can be guided visually towards ranges of parameter values with much influence on the extracted features. For this purpose, we propose three dedicated overview visualizations in Section 4. A novel triangular subrange correlation view (TSCV) provides an overview of the computed correlation values for all parameter subranges. A parallel correlation strength view (PCSV) shows the aggregated correlation strength per label and feature accumulated over all subranges. A tabular correlation deviation view (TCDV) visualizes the deviation of average and min/max correlation strengths.

In Section 5, we combine our new views with an existing visualization of parameterizations and corresponding segmentation outputs [RLK*15]. To enable analysts to study selected parameter dependencies in more detail, we will introduce a new mechanism to emphasize the scope of impact of a dependency based on the underlying correlations. In addition to linking and highlighting techniques, we propose a special sorting strategy tailored to deal with segmented time series.

Section 6 demonstrates how our approach can be applied in the context of activity recognition. The studied segmentation algorithm has five parameters and is supposed to label segments according to 16 human activities. We present key findings about parameter dependencies in a dataset of about 750 alternative segmentations generated by the algorithm with different parameter configurations.

To evaluate our approach, we conducted an informal interview with an expert from the field of activity recognition. The collected feedback is presented in Section 7. We close with a discussion in Section 8 and a brief conclusion in Section 9.

2. Background and Related Work

Before our approach is explained in detail, this section will introduce the necessary background information and reflect on related research and open problems.

2.1. Data and tasks

Our primary interest regards the visual analysis of dependencies between parameters and segmentations. The assumption is that a segmentation algorithm has a set of parameters \( P \). To be able to run, the segmentation algorithm requires a parameter configuration, or short parameterization \( p : P \rightarrow \mathbb{P} \), which assigns a concrete parameter value for each parameter. For the sake of simplicity, we assume that all parameter values are from the same parameter domain \( \mathbb{P} \).

Using a time series and a parameter configuration as input, the segmentation algorithm generates a segmented time series, or short segmentation, as output. A segmentation is a sequence of segments where each segment is characterized by a label \( l \in L \) and a duration \( d \in \mathbb{R} \). The position of a segment in time corresponds to the accumulated duration of its preceding segments. There are no gaps between segments.

For a parameter space analysis, a whole set of parameterizations \( \mathcal{P} = \{ p_1, p_2, \ldots, p_n \} \) must be configured. We use \( p_i(p) \) to denote the value of parameter \( p \in P \) in the \( i \)th parameterization. Executing the segmentation algorithm for all parameterizations in \( \mathcal{P} \) yields a whole set of segmentations \( \mathcal{S} = \{ s_1, s_2, \ldots, s_n \} \). That is, for each parameterization \( p_i \), there is a corresponding segmentation \( s_i \).

Because time-series segmentation is usually a computationally expensive procedure, testing each and every combination of parameter values exhaustively is typically impossible. Instead, the analysis usually starts with a limited, but well-chosen \( \mathcal{P} \) [SHB*14]. In subsequent steps, \( \mathcal{P} \) can be refined based on knowledge gained during the analysis [BBB*18].

For the refinement, it is helpful to know which parameters have a high influence. A parameter has a high influence if varying its value produces segmentations with different properties. In contrast, if segmentation properties are similar even for very different parameter values, one can assume that the parameter has only little influence. By adjusting influential parameters first and omitting parameters with less influence, the number of parameterizations to be investigated can be reduced.

Finding out that a parameter has an effect is but one part of the analysis. It is also important to understand how a parameter influences the results. A parameter might regulate different properties of the segmentation, including the number of segments in the sequence, the duration of individual segments or the assigned labels. Analysts need to identify the affected properties to judge whether changing a parameter leads to desired results or not. Further, they have to know which parameters can be adjusted to control a certain aspect of the segmentation. This involves constraining parameter values selectively to ranges for which a desired segmentation behaviour is obtained.

In summary, the analysis of parameter dependencies as considered in this work is centred around two key questions:

• What is influenced?—Starting with a particular parameter from \( P \), the goal is to understand what aspects of the segmentations \( \mathcal{S} \) are influenced by it, or short \( P \rightarrow \mathcal{S} \).
• Who is influencer?—Starting with certain segments in the segmentation data \( \mathcal{S} \), the goal is to understand which parameters in \( P \) exert influence on them, or short \( \mathcal{S} \rightarrow P \).

Both questions include unknowns that demand an interactive visual analysis supported by automated computations. For example, it depends on the concrete analysis context which properties of the segmentation and which parameter subareas are considered relevant.
and interesting. Moreover, there is no clear threshold of what is considered a strong dependency or a substantial influence.

Therefore, our goal is to support a human-in-the-loop analysis with appropriate visual, interactive and analytical means. Ultimately, a good understanding of parameter dependencies will allow domain experts to find interesting new parameterization, investigate different segmentation behaviours and predict new segmentation results.

2.2. Application scenario

Activity recognition is a field where understanding parameter influence on the segmentation outcome, i.e. the recognized activities, is crucial [KNY*14]. This makes activity recognition a suitable application scenario to illustrate our approach. Next, we briefly outline the application background and sketch some analytical problems that an expert in activity recognition might need to deal with.

In our case, the aim of the activity recognition is to automatically identify the activities of an observed human subject. In total 16 different activities are to be recognized correctly, including cutting ingredients, cooking, eating, cleaning up and putting away utensils. The raw time-series data are collected with a sensor suit with multiple inertial measurement units attached to the lower legs, lower arms and upper back of the subject.

The sensor data are processed by an activity recognition algorithm to produce segmented and labelled time series. Five parameters are involved in the segmentation process. They steer the reduction of the sensor data (‘Obs’), the quality of the applied state model (‘L1’), the applied filters to reduce the state space (‘Mode’), the temporal resolution (‘Timing’) and a limit value for the detection of activities (‘Dist’). The individual parameters have between two and six different values. Here, we use a dataset with about 750 parameterizations and corresponding segmentations.

The task of domain experts is to find out which parameterizations lead to good, in the sense of accurate, recognition results. At the same time, they have to keep an eye on the computational costs involved in generating the good results. That said, the domain experts have to go through many alternatives with potentially conflicting options to improve the activity recognition. Two such analytical problems are sketched below:

• Several segmentations have been created with a coarse sampling of the parameter space. A comparison with video-taped ground truth, however, shows that some activities have a low recognition accuracy. In this case, the aim of the domain expert is to improve the detection of said activities. This requires finding out which parameter values lead to better detection accuracy without compromising the accuracy for other activities.
• After testing many parameterizations, an overall satisfying recognition rate could be achieved. However, the high value of a particular parameter leads to long runtime of the activity recognition prohibiting fast response times. Now the aim is to find out how far the value of said parameter can be lowered without disturbing the already achieved recognition accuracy too much.

These are but two examples of typical domain questions. Both of them require a good understanding of the influence of parameters on the activity recognition. Supporting domain experts in developing such a good understanding is our goal.

2.3. Previous research and open problems

Previous research has proposed different ways to tackle the challenge of understanding parameter dependencies. Automatic methods employ purely computational procedures. For example, sensitivity analysis is based on automated calculations that estimate how the response of a system is linked to changes of its inputs [Han94, HJSS06]. A sensitivity analysis can also inform the prediction of uncertainties in the produced results [XG08]. Furthermore, different parameter space partitioning methods [STV17, PKNM06, Bis06] and sampling strategies [HJSS06] can be applied to identify distinct regions within the parameter space.

Purely automatic methods, however, can only be applied if the produced outputs are not too complex, if potentially interesting properties of the outputs are known and if the distinct regions can be defined in a mathematical way. Yet, in many application scenarios, this is not the case. Therefore, human expertise needs to be involved in the analysis process. This is the reason why automatic methods are often combined with interactive visualization to conduct a more dynamic exploratory analysis.

Sedlmair et al. provide a systematic overview of approaches to interactive visual parameter space analysis [SHB*14]. To identify parameter influence, sensitivity analysis [BPFG18] or trend analysis [GWR09] can be combined with visualization and interaction to focus early on relevant parts of the data. Affected properties can be revealed by visually linking the parameters to clustered algorithm output [BM10] or measured deviations from a reference model [USKD12]. The exploration of regions with distinct algorithm behaviour can be facilitated by incorporating user feedback based on which partitioning criteria can be refined incrementally [BSM*13, MP13, BPFG18]. Interactive visual approaches to parameter space analysis have been successfully applied in different contexts, including image processing [PBCR11, TWSM*11], analysis of large volumetric datasets [BAAK13, BM10, AHRG10] and fisheries management [BMPM12].

There is also previous research on understanding parameter dependencies specifically for time-series segmentation [LRHS14, BDB*16]. These works primarily focus on providing new visualization techniques. In particular, our approach draws upon the idea of visualizing different parameterizations and corresponding segmentation results side by side to allow analysts to visually evaluate parameter dependencies [RLK*15]. Yet, the means provided to support the search for and the confirmation of parameter dependencies is limited in these previous works. Analysts still have to do a lot of guesswork when determining which parameterizations influence which segmentation properties, simply because the search space can be considerably large.

In fact, the size of the search space depends on the segmentation algorithm, the generated segmentation output and the analyst’s interests, more concretely on the number of parameters and individual
parameter values, on the number of labels being assigned and on the number of segmentation properties being of interest. If we assume, for the purpose of illustration, that there are only half a dozen of each, parameters, individual parameter values, labels and segmentation properties, the search for dependencies has to cover already thousands of candidates. Inspecting all of them by hand would be very cumbersome. Therefore, the analyst should be supported in the search for parameter dependencies.

In summary, previous research already describes useful approaches to visual parameter space analysis. However, conducting a comprehensive analysis still involves much manual work. Our goal is to support the analysis and to make it easier to identify influential parameters, determine affected parts of the segmentation and pinpoint parameter ranges that show interesting behaviour. To this end, we utilize correlation calculations based on features as introduced next.

3. Feature-Based Correlation Calculation

The aim of calculating correlations is to find parameter dependencies automatically. The obtained results are then used to drive interactive visualizations that help analysts to understand the big picture of the dependencies existing in the data and also to study individual dependencies in local subranges.

The idea behind our correlation-driven parameter space analysis follows the same principle as the manual search: We seek matches between changes in parameter values and resulting changes in the sequences of labelled segments. However, the latter changes are difficult to quantify because segmentations can vary in different ways. The number of segments can vary, the segments can shift in time and the segments can have different labels. Moreover, the amount of data to be processed is large, rendering a comprehensive analysis including all details infeasible.

What is needed in the first place is a way to make segmentations, more precisely the properties of segmentations, mathematically easier to investigate and compare. For this purpose, we propose to define features that characterize a segmentation.

3.1. Defining features

A feature $f : S \rightarrow \mathbb{R}$ abstracts a segmentation to a single quantitative value and as such serves as a compact formal description of a relevant segmentation property.

A feature can be label-dependent or label-agnostic. Label-dependent features are defined on a per-label basis. We consider the number of segments cnt, with a particular label $l \in L$, the average duration of these segments dur, and their average temporal position pos. With these features, it is possible to tell if many or only a few segments are tagged with certain labels, if the segments are short or long and if they occur more in the beginning of the observed time span or in the end. An additional label-agnostic feature counts the number of unique labels appearing in a segmentation to quantify the label diversity div. Future work could include additional features if the application domain demands so.

Together, the features form a feature set $F$, which is specific to the labels $L = \{l_1, l_2, \ldots, l_n\}$ of the analysis problem at hand:

$$F = \{\text{cnt}_{l_1}, \text{cnt}_{l_2}, \ldots, \text{cnt}_{l_n},$$

$$\text{dur}_{l_1}, \text{dur}_{l_2}, \ldots, \text{dur}_{l_n},$$

$$\text{pos}_{l_1}, \text{pos}_{l_2}, \ldots, \text{pos}_{l_n},$$

$$\text{div}\}.$$}

Computing the features for all segmentations in $S$ yields a set of feature descriptors $F = \{f_1, f_2, \ldots, f_m\}$. For each segmentation $s_i \in S$, there is a corresponding feature descriptor $f_i \in F$. We use $f_i(f)$ to denote the value of feature $f \in F$ in the $i$th feature descriptor.

Each individual feature descriptor captures the characteristic properties of its associated segmentation. The key benefit of using features is that the problem of analysing complex size-varying sequences of labelled segments is reduced to analyse a fixed number of quantitative feature values. The amount of data is greatly reduced, which makes it possible to calculate correlations for several thousands of parameterizations and feature descriptors in a comparatively short time.

3.2. Calculating correlations of parameters and features

The literature describes different ways to calculate correlations. We use the Pearson correlation to compute the correlation coefficient $r_{p,f}$ between a parameter $p \in P$ and a feature $f \in F$ as:

$$r_{p,f} = \frac{\sum_{i=1}^{m}(p_i - \bar{p})(f_i - \bar{f})}{\sqrt{\sum_{i=1}^{m}(p_i - \bar{p})^2 \sum_{i=1}^{m}(f_i - \bar{f})^2}}$$

where $p_i$ and $f_i$ are the parameter value and feature value of the $i$th parameterization and feature descriptor, respectively. The $\bar{p}$ and $\bar{f}$ are the averages over all indices $1 \leq i \leq n$.

Correlation coefficients close to zero indicate that a parameter has no influence, whereas values close to 1 or −1 suggest a high influence. For example, if the calculation yields a correlation coefficient of −0.8 between parameter $p_i$ and the feature $\text{dur}_{l_5}$, it is likely that the fifth parameter has a strong influence on the duration of segments labelled with $l_5$. The negative correlation value signifies that increasing the parameter value will shorten the segments.

The aforementioned correlation computation iterates over all available parameterizations and feature descriptors. However, such a global calculation will capture only the overall influence of parameters. To detect local parameter influence, the correlation calculation must be performed on parameter subranges. A correlation calculation concerning a subrange $[a, b]$ of parameter $p \in P$ no longer iterates over all indices, but only over a subset of indices. As a consequence, the sums in the above correlation formula become $\sum_{i \in I_{p(a,b)}}$, where the index subset

$$I_{p(a,b)} = \{i \mid a \leq p_i \leq b\}$$

contains only the $i$ whose corresponding parameterization exhibits a value of parameter $p$ in the range $[a, b]$. 
A comprehensive search for local dependencies requires calculating the correlation for multiple subranges. This can be done in a bottom-up procedure. It starts with many rather small subranges, where \( a \simeq b \). In subsequent steps, fewer but increasingly wider subranges are created by merging the ranges of previous steps until a parameter’s entire value range is covered. With this procedure, each combination of upper and lower boundaries for parameter values is considered and each parameter subrange gets its individual correlation value \( r_{[a,b],f} \).

To perform the necessary calculations efficiently, partial sums of the Pearson correlation for smaller subranges are reused for the larger subranges. This leads to a runtime complexity of \( O(n^2) \), with \( n \) being the number of parameterizations. In practice, processing thousand parameterizations takes less than a second, which allows us to calculate correlations live during an analysis session. This opens up opportunities for integrating other automated methods that generate new parameterizations and corresponding segmentations on the fly.

As the result of the correlation calculation, we know which parameters correlate with which features and thus have an impact on the properties of the time-series segmentation. We know this not only globally, but also locally for different parts of the parameter space. The next step is to visualize the correlations.

### 4. Correlation Visualization

Understanding parameter dependencies on the basis of the calculated correlations involves studying three important aspects:

- The calculated correlation values \( r_{[a,b],f} \) are useful for estimating how a certain subrange of a particular parameter affects a particular feature.
- With the help of absolute correlation strengths \( |r_{[a,b],f}| \), one can distinguish influential parameters from non-influential parameters.
- Looking at deviations of the maximum and minimum correlation strength from the average correlation strength allows analysts to evaluate if the influence is similar across all subranges, or if the subranges exhibit diverging correlation behaviour.

Next, we introduce three dedicated visualization views, one for each of the three aspects just mentioned.

#### 4.1. Overview of correlation values of subranges

The correlation values for all subranges are visualized in a novel TSCV. The visualization is inspired by the triangular model, a diagrammatic representations for intervals [Kul06, QDV+12].

The TSCV consists of a triangular arrangement of boxes, each representing the correlation value \( r_{[a,b],f} \) for a particular subrange \([a, b]\). Figure 1 shows an example with the correlation values for the parameter \( p = \text{Mode} \) and the feature \( f = \text{cntCUT} \). The subrange boundaries are encoded by varying the box positions. The position for the highlighted box of subrange \([0, 2]\), is determined by constructing two oblique lines. One emanates from 0 at the horizontal axis with an angle \( \alpha \), the other starts at 2 and takes an angle of \( 180 - \alpha \). The box is placed where the two lines meet. The angle \( \alpha \) can by varied to control the aspect ration of the view.

With this layout schema, subranges with a wider extent are placed higher up in the view, the largest possible subrange covering the entire value range is at the top. Narrow subranges are located at the bottom. Subranges with a small lower boundary are located to the left, while subranges with a high upper boundary can be seen to the right. To make the layout procedure more clear, two legs connect each box to its lower and upper boundary at the horizontal axis. Note that these legs do not imply any hierarchical relationship or dependencies between the individual subranges.

The actual correlation value per subrange is encoded in the colour of the box using a diverging colour scale from ColorBrewer [HB03]. The example in Figure 1 shows correlation values differing greatly across the subranges. The faint blue colour of the upper box indicates a weak negative correlation when considering the entire parameter range \([0, 3]\). The saturated red colours of the lower right boxes indicate a strong positive correlation when parameter values greater than two are involved.

With the TSCV, it is easy to see if a parameter is correlated with a feature only for short parameter ranges or for longer ones. This allows the analyst to judge a high correlation value as merely an outlier or to conclude that a parameter influences the behaviour of the segmentation algorithm in general.

In order to get a broader view on parameter dependencies, we extend the basic TSCV in two ways. Firstly, we replace the flat box representing a single feature with a matrix representing the correlation values of all features simultaneously. Secondly, we show multiple TSCVs side by side, each representing the correlations of a different parameter. Figure 2 shows such an extended visualization for three parameters. One can see that the correlation values of the parameter ‘Obs’ are high and evenly distributed over all features, while the correlation values of the parameter ‘L1’ are mostly negligible with a few exceptions in the narrow parameter ranges. The
Figure 2: The extended TSCV visualizes correlation values for all features in the cells of small matrices. Several parameters are shown side by side to support comparison. Due to different numbers of values per parameter, the individual TSCVs contain different numbers of boxes.

Figure 3: The parallel correlation strength view (PCSV) visualizes the average correlation strength for parameters, features and labels. In the shown example, the influence of the parameter ‘Obs’ on the features cnt and \textit{dur} for all labels can be compared.

fluctuations in the correlation distribution for the parameter ‘Mode’ indicate its uneven influence.

The TSCV provides a good initial overview of the correlation between parameters and features. It is well suited to compare in which subranges certain parameters take effect. To a limited degree, it is also possible to see which features are affected. However, interpreting the view becomes more difficult when the number of visualized correlation values increases. Therefore, we visualize aggregated correlation strengths in a second dedicated view.

4.2. Overview of correlation strength

To help analysts to better understand which features are influenced by which parameters, we integrate a PCSV. The PCSV is a parallel-coordinates-style visualization designed to visually communicate four aspects: the features (\textit{cnt}, \textit{dur}, \textit{pos}, \textit{div}) being influenced, the labels \textit{L} being influenced, the parameters \textit{P} exerting the influence and the aggregated correlation strength $|r_{p[a,b],f}|$.

Because the labels can be numerous, they are represented as the parallel axes of the view. The strength of influence is represented as polylines connecting the axes. As there are only a few parameters and features, the polylines’ colour and dash pattern can be varied to encode these two aspects. To reduce clutter of polylines, the PCSV shows them selectively on demand.

Note that we consider the absolute correlation strength, rather than the original correlation value, because we are now primarily interested in how strong the influence is. It is further worth mentioning that the visual encoding of the PCSV disentangles the label-dependent features into a feature part and a label part. This was a deliberate design decision to limit the number of parallel axes.

Studying the PCSV, analysts can estimate and compare the overall influence of each parameter. For example, to evaluate whether a parameter affects the number of segments (\textit{cnt}) or the duration of segments (\textit{dur}), or both, one only needs to compare the polylines with different dash patterns as illustrated in Figure 3.

Comparing the strength of influence of different parameters on the same feature is possible by looking at lines with different colours (but the same dash pattern). The comparison can be refined further with regard to the labels. For example, in Figure 4, the influence of parameter ‘Obs’ in blue is generally stronger than the influence of ‘Mode’ and ‘L1’ in red and orange. Yet, there are three labels (CUT, FILL and MOVE) onto which parameter ‘Mode’ exerts comparable or even greater influence than parameter ‘Obs’ does. This means that if analysts have to tweak the properties of segments with these particular labels, they should try adjusting the values of both parameters.
The tabular deviation view (TCDV) offers an overview of the deviations of correlation strength. Red and green bars visualize the deviation of the minimum and maximum correlation strength from the average, respectively.

4.3. Overview of correlation deviations

Understanding different patterns of correlation distributions can be important when analysts want to re-parameterize the segmentation algorithm and predict the outcome. For example, if a parameter’s correlation values are consistent throughout all sub-ranges, the segmentation outcome for new parameterizations can be safely extrapolated. On the other hand, sub-ranges with diverging correlation values may indicate a varying influence on the segmentation behaviour. In such cases, analysts might be interested in finding the parameter values at which the influence changes.

While the PCSV already visualizes the overall parameter influence by showing the average correlation strength, it completely abstracts from the individual sub-ranges. The TSCV is capable of visualizing the correlation distribution, but it cannot show all correlation values at once. Therefore, we introduce a third view, the TCDV, which aims to support the identification of combinations of parameters, features and labels for which the correlation strength in the underlying sub-ranges deviates substantially from the average.

The TCDV consists of a tabular display of rows and columns as shown in Figure 5. The rows are arranged in groups. There is one group per parameter, and each group consists of rows corresponding to the different features (cnt, dur, pos, div). As for the PCSV, features and labels are disentangled. The labels correspond to the columns of the TCDV.

Each cell in the TCDV visualizes the correlation deviation among the sub-ranges associated with a parameter-feature-label triple. Per cell, there are two bars extending from the centre, a red bar to the left and a green bar to the right. The red and green bars represent the normalized deviation of the minimum and maximum correlation strength, respectively, from the average correlation strength. Cells with wide dark bars indicate that the correlation strength varies greatly, whereas narrow and bright bars stand for little variation. This visual encoding guides analysts to combinations of parameter, feature and label that exhibit heterogeneous correlation values in the underlying sub-ranges.

In summary, the three proposed visualizations support the assessment and comparison of the influence of parameters on the different features and labels. This makes it possible to distinguish influential from less influential parameters and to determine the influenced properties of the segmentations. Moreover, the correlation distribution in different parameter sub-ranges can be evaluated in order to get a more detailed view of the peculiarities of the studied algorithm’s segmentation behaviour.

5. Correlation-Driven Interactive Exploration

The three views introduced before provide first overviews of the calculated correlations between parameters \( P \) and features \( F \). Yet, developing a comprehensive understanding of parameter dependencies additionally requires useful interaction facilities and a visualization of the segmented time series \( S \). Next, we describe how coupling the previously introduced views with a visualization of \( S \) and corresponding interactions can support different exploration strategies for the analysis of parameter dependencies.

5.1. View linking

To facilitate the exploration of correlations, the TSCV, PCSV and TCDV are interlinked and support the standard way of dynamically filtering the views based on parameters, features or labels. Moreover, selecting a parameter-feature pair, for example, by clicking a data point in the PCSV or a cell in the TCDV, triggers two linked effects. Firstly, the TSCV shows the detailed correlation values of the selected pair. In this way, the analyst can see how the average correlation strengths in the PCSV or the deviations in the TCDV are composed by the underlying sub-ranges. Secondly, in the PCSV, the polyline corresponding to the selected pair is replaced by a band that spans from the minimal to the maximal correlation strength. This allows the analyst to see the spread of the correlation strength of the different labels.

Figure 6 illustrates how the linked interaction can be utilized. In the visualization to the left, the wide bars under the cursor indicate a heterogeneous influence of the parameter ‘Mode’ on the feature ‘cnt’ (top row of the TCDV). Clicking the corresponding cell results in the visualization shown to the right. The dimming in the other rows of the TCDV makes clear what the current focus of the exploration is. The rather wide red band that has appeared in the PCSV (top) confirms that the maximum and minimum correlation strength deviates strongly from the average. Further, the TSCV (bottom) shows that the correlation is strong only towards the end of the parameter’s value range. To determine the exact threshold, new parameterizations would need to be created, and corresponding segmentations and correlations be calculated. The TSCV suggests the parameter range \([1.0, 2.0]\) to be suitable for these calculations.
5.2. Integrating correlation and segmentation visualization

The linked overview visualizations facilitate the exploration of correlations between parameters \( P \) and features \( F \). Yet, analysts also need to understand how these correlations connect to the concrete segmentations. Therefore, the presented overviews are designed to work in concert with a visualization of the actual segmented time series \( S \). How this visualization looks like and how it can be enhanced based on the calculated correlations will be explained in the following.

**Visualizing segmentations.** We utilize an existing tabular visualization of parameter-dependent segmented time series [RLK*15]. As illustrated in Figure 7(1)(1), the visualization consists of a vertical stack of coloured rows, each representing a parameterization \( p_i \in P \) and its corresponding segmentation \( s_i \in S \). On the left side, the parameterizations are depicted by showing the combinations of applied parameter values. Dark and bright colours for the individual parameters encode low and high parameter values, respectively. The segmentations are shown on the right side. Each row visualizes the sequence of labelled segments from left to right, where colours indicate the label associated with each segment. The duration of a segment and its temporal position within the time series is encoded in the horizontal position and the length of the coloured band.

This visualization provides a valuable overview of \( P \) and \( S \). However, to visually reveal dependencies between parameters and segments, the rows of the visualization must be sorted properly. So far, this sorting had to be done manually by the analyst in a costly trial-and-error procedure. Here, we utilize the calculated correlations to enhance the visualization.

**Emphasizing and sorting.** Our goal is to make parameter dependencies easier to grasp and understand by emphasizing those parts of the data that correlate substantially. For this purpose, we combine automatic highlighting and sorting.

To emphasize a particular parameter dependency indicated by a high correlation strength, three aspects need to be communicated: the influencing parameter \( p \in P \), its subrange of influence \([a, b]\) and the labelled segments of the segmentations \( S' \subseteq S \) being affected. The corresponding emphasis procedure is as follows.

Firstly, parameters and segments not involved in the dependency being studied are dimmed. Moreover, the borders of affected segments are accentuated and the regions around them are brightened to produce a highlighting effect. As illustrated in Figure 7(2)(2), this makes the influencing parameter and the affected segments easier to see.

The second step is to sort the rows of the visualization. The sorting uses a primary and a secondary sorting condition. The primary condition is based on the parameter values. It results in the rows being sorted according to \( p \), more precisely according to \( p_i(p) \). Figure 7(3)(3) suggests that this can already help the analyst to develop an initial idea of the depicted parameter dependency. Yet, segmentations \( s_i \) that are associated with parameterizations that have the same value \( p_i(p) = \text{const} \) are still in arbitrary order, which can obscure the actual nature of the dependency.

Therefore, if the primary condition cannot resolve order due to equal parameter values, the secondary sorting condition arranges rows according to segmentation similarity. This secondary sorting requires two additional calculations, as illustrated in Figure 8:

1. Compute pairwise edit distance between segmentations to measure their similarity.
2. Find a sorting order with minimal edit distance to place similar rows next to each other.

© 2019 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd
Figure 7: Steps to emphasize a correlation between an involved parameter and segments with the label. (1) Parameter and segmentation data without highlights and unsorted rows. (2) Highlights are applied to emphasize the involved parameter and segments. (3) Highlights of the parameter range of influence are added and rows are sorted by the values of the involved parameter. (4) Rows within the same parameter subrange are sorted by similarity.

Figure 8: Procedure for local sorting based on similarity. Firstly, for each pair of segmentations, the edit distance is computed. Then, a travelling salesman problem (TSP) is solved to determine the sorting order.

Firstly, the edit distance $\Delta(s_i, s_j)$ between any pair of segmentations $s_i, s_j \in S$, is calculated. The edit distance counts the number of edit operations that would be necessary to transform $s_i$ into $s_j$. The edit distance considers only the segments that are involved in the parameter dependency being studied.

Secondly, the sorting order is determined by minimizing the sum of edit distances between adjacent segmentations. This requires solving a travelling salesman problem (TSP), where the $s_i$ are the nodes and the calculated edit distances define the edge weights. The TSP’s solution is a shortest path whose nodes define the sorting order.

As can be seen in Figure 7(4)(4), the resulting visual representation is now easier to interpret. Thanks to the similarity-based sorting, the affected segments vary only a little from one row to the next. The considerable fluctuations being evident in the unsorted visualization in Figure 7(3)(3) are no longer an issue.

© 2019 The Authors. Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd
What is influenced?

For a

analysis, the goal is who-is-influencer

2019 The Authors.

An analysis with a

published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd

Who is influencer?

per-

The
c. eichner et al. / making parameter dependencies visually understandable

parameters, features, labels, and subranges. To support such a comprehensive analysis, we designed a user interface that glues together the aforementioned analytic, visual and interactive techniques.

Figure 9 shows the part of the interface that is used to dynamically filter all computed correlations with regard to certain interest. Its most important element is the list of calculated correlations. Each item in the list contains the correlation value for a particular combination of parameter, feature, label and subrange. The list is sorted by the correlation strength, placing the most important entries at the top. When a correlation is selected, all visualizations are automatically configured to show and/or emphasize the corresponding parameter, feature, label, and subrange.

The interface further contains controls to filter with respect to parameters, labels, features (top) and subranges (bottom). The colour-coding in the parameter filter and the label filter visualizes the maximal correlation strength. This makes it easy to quickly narrow the analysis down to key influencing parameters and key influenced labels.

Figure 10 shows all developed components together. The interface uses two windows for the correlation visualization (left) and the data visualization (right). Both windows are linked by design, but can also be used separately on demand.

5.4. Exploration strategies

The interface in Figure 10 enables analysts to study parameter dependencies with respect to the two questions described in Section 2.1: What is influenced? and Who is influencer? The two questions naturally lead to two exploration strategies. To explore what is influenced, the visualizations and interface components are utilized from left to right. To find out who is influencer, the exploration works in the opposite direction from right to left.

What is influenced? An analysis with a what-is-influenced perspective starts with the overview visualizations TSCV, PCSV and TCDV depicted in Figure 10(a)(a). The three views are used to identify the influential parameters and strong correlations to certain features. Each view can be configured manually to display only correlations of selected parameters or features (Figure 10bb).

Based on the gained insight, the analyst sets filters to focus on smaller sets of correlations, for example, to investigate strong correlations in certain subranges (Figure 10cc). To support selecting interesting parameters, the filter controls highlight parameters and labels with strong correlations. Thanks to the linking between views, filters can also be defined directly from the overview visualizations, for example, by selecting polylines in the PCSV or cells in the TCDV.

With the filters applied, the correlation list shows only those correlations that match the filter criteria. Now, the analyst can inspect the strongest correlations by simply clicking them. This causes the visualization of the segmentations to highlight the involved segments and sort the rows accordingly (Figure 10dd).

By inspecting one correlation after the other, the analyst can develop a detailed understanding of how parameters influence the segmentation. This involves systematically changing the filters to direct the analysis to other parameters, features and labels, until the important relationships in the data have been considered.

Who is influencer? For a who-is-influencer analysis, the goal is to find the parameters and corresponding subranges that influence selected segments. So, the analyst considers the visualization of the segmented time series first and pinpoints segments of interest. Segments with a certain labels can be selected with the interface (Figure 10ee). Alternatively, if the analyst spots an interesting pattern in the visualization (Figure 10dd), segments can be selected there directly.

The correlation list (Figure 10cc) is then reordered to show correlations that match the selection best and have high correlation...
values. The parameters involved in the correlations at the top of the list have good chances of influencing the segments of interest. If some types of features (e.g. \textit{cnt} or \textit{dur}) appear more frequently in the list, the chances are high that changes to these segment properties initially sparked the interest for the selected part of the data.

It is possible that the found correlations in the list are only strong in a very small subrange or that other parameters are similarly influential. A complete analysis therefore requires to put the observed correlations into an overall context. For that, the visualizations of the correlation values (Figure 10aa) are used to compare parameter influence and to investigate whether the found correlation values are merely outliers or also exist over a larger parameter value range. For this purpose, the correlation visualizations are configured (Figure 10bb) to only incorporate correlations that are similar to the correlations of the filtered list.

In summary, the described exploration strategies utilize the developed techniques to facilitate the analysis of parameter dependencies of time-series segmentation. Analysts can determine the effects of the most influential parameters on the segmentation and also link observations in sub-areas of the segmented time series to certain parameter values. Next, we illustrate how our approach can be applied in the context of activity recognition.

6. Application to Activity Recognition
In this section, we look in detail at the data from the activity recognition scenario described in Section 2.2. As mentioned before, the activity recognition depends on five parameters and we are dealing with 750 parameterizations and corresponding segmentations.

6.1. Known dependencies
Previous analyses [RLK*15, BBB*18] of these data revealed, for example, that the parameter ‘Obs’ influences the length of segments labelled EAT and COOK, which are shown as red and green segments in Figure 10 and also in Figure 7. Further, the differentiability of some activities, for example, EAT and DRINK, is influenced by the parameter ‘Mode’. As expected, a clear separation of activities becomes more difficult, the more the original sensor data are reduced (‘Obs’) and the simpler the model is (‘Mode’). This could be confirmed visually for long segments (e.g. for EAT and COOK), but it remained difficult to see the influence on short segments (e.g. for MOVE or PUT).

6.2. Findings with correlation calculations
Our goal was to replicate the previous findings and to test if we can generate interesting new findings with the correlation-driven approach.

In a first step, 51 different features were calculated for each segmentation, and 2346 correlation values were computed to link the features to 46 parameter subranges. The feature extraction took about 0.15 s and the correlation calculation took 0.05 s on a regular desktop PC with a 2.9 GHz CPU.

With the computed correlations, some of the previously made findings could be confirmed easily. For example, the mentioned relationship between the parameter ‘Obs’ and the segment length \textit{dur} of the activity EAT has a correlation value of $-0.86$. The correlation between ‘Obs’ and COOK is $-0.80$. These quite strong correlations are among the 10 strongest correlations in the data. The negative sign confirms that a reduced sensor resolution leads to longer segments.
Although this is not a new finding in itself, it should be mentioned that making this observation is easily possible just by referring to the strongest correlations in the user interface in Figure 9.

With the help of the new overview visualizations, it now additionally becomes clear that the influence is not limited to these longer activities. The influence of the parameter ‘Obs’ is generally quite strong and also the somewhat weaker influence of ‘Mode’ on the number of segments is clearly visible from the representation of the correlation strength in Figure 4. The influence of ‘Obs’ is also present for short activities. For example, the number of segments \( c s i_{P U T} \)
, where PUT denotes the short activity of putting down an item, is correlated to the parameter ‘Obs’ with a factor of 0.77 as listed in the interface in Figures 9 and 10. This implies that with increasing values for ‘Obs’, the previously long segments not only become shorter, but also new shorter segments are recognized more often.

Figure 11 highlights the dependency between the parameter ‘Obs’ and the label PUT. It is clearly visible that a lot more green segments are recognized in the top rows, for high values of ‘Obs’, than at the bottom, for low values of ‘Obs’. Finding and evaluating this relatively strong dependency without our correlation-driven approach would be quite difficult, because short segments are otherwise not so easy to see and compare. In this regard, our approach can simplify the interaction to find known relationships and also to track down previously unknown relationships.

Finally, the extended TSCV in Figure 2 makes it possible to understand how the parameters exert their influence differently at the level of subranges. The parameter ‘Mode’ shows slightly stronger negative correlations only over long-value ranges (bluish cells in the matrices at the top). The parameter ‘Obs’ has relatively strong influence that is evenly distributed across all subranges and labels (many red and saturated cells in most matrices). In contrast, the effect of ‘L1’ is limited to very small subranges (red cells only in the matrices at the bottom).

7. Expert Feedback

To conduct a first test of our techniques, we carried out an expert feedback session. Our aim was to collect informal feedback for the plausibility of the features and correlation calculations, the usefulness of the visualizations of correlations and data and the utility of the interactive exploration facilities. To this end, we recruited an academic expert with experience in time-series segmentation and activity recognition. The expert has previous knowledge on segmentation algorithms in the context of ambient assisted living and on classification methods for driver behaviour in context of automotive applications. That said, he is well acquainted with the tasks and the challenges associated with evaluating the outcome of parameter-dependent time-series segmentation. The expert was not involved in the development of the approach presented in this paper. The dataset described in Section 6 was used during the feedback session.

The feedback session started with a description of the background, the goals and the tasks to be supported by our approach. In the first part of the interview, the expert tried to solve the tasks without any support from correlation-driven methods. Afterwards, we presented our approach, introducing the idea of correlation calculations based on features, the three different correlation visualizations and the utilization of correlations for emphasizing relationships in the visualization of the segmented time series. Each visualization design was described and an example for a possible interesting observation was given. Afterwards, the expert got time to look at the data himself and was asked to think aloud about how he was using the provided techniques. The interviewer further asked questions related to the described tasks, the usefulness of the approach and possible improvements. An assistant produced a written transcript of the feedback session. The session took about 1 h.

7.1. Feedback regarding standard approach

In the first phase of the session, the analyst worked only with the parameterizations and the segmentation data visualized side by side as in Figure 7(1)(1). The system was configured to allow manual sorting of rows by parameter values as well as sorting by segment similarity. Highlights could be assigned manually to emphasize segments with certain labels. In this part of the interview, none of the new correlation-driven techniques were enabled to create a setting that is similar to the state of previous work.

The initial questioning aimed to identify influential parameters by asking for parameters being responsible for changes in the segments. After about 5 min of trying different sorting strategies for the rows, the expert was able to identify the parameter ‘Obs’ as clearly influential, while he had a weak suspicion that the parameter ‘Mode’ could also have an influence. After confirming this general observation, we asked whether he could identify how the parameter ‘Obs’ influences the segmentation, more precisely if increasing parameter values lead to more shorter or longer segments. Although he already knew of an existing relationship, he had clear problems describing it in detail. He stated that relationships are ‘very difficult to see from just this visualization’ of segmentations and even after manually highlighting influenced segments it was ‘still very hard to judge whether there is really a relationship, because small changes might go unnoticed’.
Only for the case shown in Figure 7, the expert was quite sure that the red segments are influenced by the parameter ‘Obs’, and he could even identify the subrange in which ‘the red segments get longer and more consistent’. However, he pointed out that identifying large segments is typically not that important. ‘The parameters of activity recognition algorithms are often primarily tweaked toward recognizing critical activities correctly, rather than all possible activities. [. . . ] Critical activities might be very short. For example, switching off the oven is critical but takes just a fraction of the recorded time. Hence, the parameters that influence these critical activities can be more important than parameters that cause changes in the majority of the data’. In the existing visualization of the segmented time series, long activities get much drawing space and therefore ‘such visualizations make it really hard to identify parameters that influence the important segments’.

7.2. Feedback regarding correlation-driven support

In the second phase of the interview, we enabled our correlation-driven analysis support. The general idea was well received by the expert: ‘This is basically what I just tried to find manually when I was looking at the data’. With regard to the recognition of critical activities, it was noted that not every conceivable feature would actually have to be considered here.

The expert paid special attention to the details of the correlation calculation in parameter subranges. After moments of deeper thinking, the idea was considered useful, potentially also for the automotive applications the expert is working on. He also noted that the approach could also be extended by considering subranges in time. This would enable the expert to narrow down the analysis to selected occurrences of (critical) activities. We think that extending our approach in the suggested way is computationally feasible by applying the presented strategy for subrange calculation to the time dimension as well. Subdividing the features by time would introduce a new dimension to the analysis: Each currently considered feature would have a whole new set of sub-features whose relationship to each parameter subrange is determined by their own correlation value. To support such an analysis, the existing correlation visualizations could be employed as top-level visualizations. For investigating the sub-features, however, extensions would need to be developed to communicate the temporal aspects.

There were further suggestions regarding the features used: ‘For example, if you have a dataset with a ground truth, you can count the number of incorrect classifications (separately for each label) and use them as new features. The correlations to these new features can then be used to directly read off the parameter values at which critical actions are more often detected incorrectly’.

7.3. Feedback regarding visualizations

In the second phase of the interview, the expert also got to see the visualizations of the calculated feature-based correlations. Starting with the PCSV, we asked him to name the most important parameters and compare their influence. After a brief examination of a visualization similar to the one depicted in Figure 4, the expert found his previous observations quickly confirmed: ‘As suspected “Obs” has clearly the highest influence, and one can only see a small influence of “L1”.’ Other parameters seem to be only moderately important.

When asked to name the labels that are similarly influenced by different parameters, the expert could quickly identify the labels CUT, FILL and MOVE, for the parameters ‘Mode’ and ‘Obs’.

The expert then configured the PCSV to show multiple kinds of features for one parameter, similar to the view shown in Figure 3. At first, he was surprised that the line of feature dur is so close to the line of cnt. He expected both to be way apart. After reminding him that we show the correlation strengths instead of the correlation values to make the influence of parameters comparable, he could understand the visualization. However, he strongly recommended to visualize the raw correlation values as well: ‘For two different parameters, the effect of increasing parameter values may be the opposite in some cases, while the same parameters may affect other features in the same direction. When searching for good values for multiple parameters, these effects have to be balanced out and therefore the correlation direction is important’. The expert also pointed out that it would be useful to hide less critical activities in the PCSV if necessary, because ‘[. . . ] otherwise the lines for one parameter could get cluttered by showing correlations from unimportant activities’.

Focusing on the TCDV, we asked the expert to find combinations of features and parameters for which the correlation values strongly diverge in different subranges. He was able to spot such patterns quickly with the TCDV and continued to investigate them further with the TSCV. Yet, he pointed out that this specific task is very uncommon: ‘[As a developer of segmentation algorithms] knowing the variance or the average is not very helpful to find good parameterizations. Instead identifying the subranges where the correlations values are the highest and lowest would be very useful’. From earlier development stages, we already had different versions of the views, which made it possible to display minimum and maximum values of correlation strength and correlation values. The expert suggested making both averages and extrema available on demand similar to the PCSV.

The TSCV was further used as a supplement to the TCDV to check the distribution of correlations across the parameter space. The expert was asked to determine the distribution of parameter influence over the subranges. The identification of evenly distributed parameter influence could be carried out quite quickly. However, cases showing a changing parameter influence like in Figure 1 raised questions: ‘Why does the correlation strength decrease from one subrange to a subrange on higher levels, and why does the direction of the correlation sometimes even flip?’ After explaining that some strong correlations only exist for particular values ranges and recapitulating how this is recognized by the performed correlation calculations, the expert acknowledged these observations. In fact, it was noted that ‘seeing all the individual correlation values is quite useful to find such details in the algorithm’s behavior’.

The expert further commented on the visual design of the TSCV. He pointed out that qualitative parameters can be relevant as well, but the TSCV relies on numeric subranges. The expert noted himself that the general idea behind the TSCV could be extended with moderate effort. The subranges would then correspond to the power set of the qualitative parameter values, instead of being the collection of all contiguous intervals. However, the representation of subranges as boxes and their positioning as currently done in the TSCV would need to be re-designed for such cases.
7.4. Feedback regarding interactive exploration

Finally, the analysis returned to the visualization of the actual segmentation data and how it can be utilized for interactive exploration. Now the focus was on using the calculated correlations to automatically emphasize dependencies and to guide the expert to the relationships in the data as found by the previously inspected correlation visualizations.

Recalling his findings in the very first part of the session, the expert was pleased to find his assumptions regarding the long red segments confirmed by the automatic highlighting. Nevertheless, even for the strong dependency he commented that 'showing the correlation value next to the highlighted segment provides additional confidence in the interpretation of dependencies. [...] Especially if changes of segments are very subtle, it is not sufficient to rely solely on a plain data visualization'.

The expert continued by exploring the data for other dependencies with high correlation values. The automatic highlighting techniques were generally well received, in particular when the shorter critical activities were investigated. The expert found that he '[...] can now see correlations that were missed earlier due to small segment length'.

While the provided filters for labels, parameters and subranges were recognized as useful for focusing on important dependencies, it was recommended to further improve on that aspect. The expert suggested to provide additional temporal filters and to link the TSCV more closely to the visualization of the segmentations.

7.5. Overall assessment

At the end of the interview, we asked for an overall assessment of the approach. In summary, the feature-based correlations and the visualizations were considered as interesting and novel techniques: ‘Correlations give me a first [in the sense of starting point for the analysis] support to find dependencies! [...] The correlations confirm what can be seen, but also help to find new dependencies that are harder to detect visually’.

The expert was quite interested in applying our approach to his own research. He mentioned that the principle of feature-based correlation calculations and the associated visualizations could easily be adapted for automatic classification of driver behaviour or handwriting recognition. To this end, only different kinds of features would need to be collected.

Finally, we received a particularly interesting suggestion to further improve our approach: ‘It would be nice if you could weight labels and parameters by their importance to the analyst to omit some details in the visualizations and provide more space to other dependencies’. This is indeed a promising idea for future work.

8. Discussion and Future Work

While our work already covers many aspects, there are still questions worth discussing and issues to be addressed in future work.

Scalability. A first question regards the practical limits of our approach in terms of scalability. For the feature extraction, it is necessary to inspect each time point for each segmentation. Therefore, the effort required for this step increases linearly with the size of the input data $k = n \cdot t$, where $n$ is the number of parameterizations and corresponding segmentations and $t$ is the number of points on the time axis. Through the feature extraction, $t$ is reduced to a constant number of features. This means that our approach can handle very long segmentations, which might even be too large to keep them completely in memory. On the other hand, the informative power of features tends to decrease with increasing segmentation length. In future work, this circumstance could be counteracted by extracting features not only globally over the complete time axis, but also locally for sub-intervals.

Because the correlation calculation uses the extracted features, the number of parameterizations $n$ is the decisive factor for all following steps, where $n$, in turn, depends on the number of parameters and their individual discretization. If, for example, the number of parameter values increases for a single parameter, $n$ also increases by the same amount.

An individual correlation calculation for one feature, one label and one selected parameter subrange can be performed in $O(n)$. In practice, this means that one correlation value can typically be calculated for several million segmentations within about 1 s. For a complete correlation calculation, $O(n^2)$ subranges must be considered, which would lead to a runtime complexity of $O(n^2)$ if implemented naively. Our calculation method reuses partial sums from already computed subranges and thus achieves a runtime complexity of $O(n^2)$.

While this is a significant improvement, it still sets limits for practical use. Based on our experience with $n = 750$ as in the activity recognition example, we can extrapolate the following runtimes. For $n = 3350$, the complete correlation calculation would take about 1 s, and for $n = 26000$, it would require about 1 min. The latter case would correspond to a scenario with, for example, five parameters and between seven and eight values per parameter. For time-critical applications, it is advisable to reduce the number of considered parameter subranges or to calculate subranges only if necessary. For data sets with even more parameterizations, pre-calculations before the actual analysis session would be possible.

Combinations of features. Currently, we calculate the correlations for each individual parameter separately. Yet, dependencies can also exist for combinations of parameters. For example, in our application example, the activities DRINK and EAT are recognized correctly only if the three parameters ‘Obs’, ‘Mode’ and ‘Dist’ are in the right proportion to each other. To find such dependencies, the correlation calculation must be extended to work on subsets of parameters, rather than on individual parameters. This significantly increases the search space, and thus, the computational costs and the effort required for interactive visual exploration. These challenges need to be addressed in future work.

Ease of use. Overall, our approach combines several sophisticated components to support a complex analysis task. While some visual components are easier to understand (e.g. PCSV or lists in the interface), others might require some training to make the most
of them. In particular, the triangular design of the TSCV required more explanations in the expert interview and also in discussions with colleagues. Yet, once we clarified that the view is meant to show subranges of parameter values (and not some hierarchical structure as in node-link diagrams), most people quickly understood how to interpret the TSCV.

In general, our approach is not designed for casual use. Studying parameter dependencies of time-series segmentation is a complex task to be carried out by experts. Although our approach makes the task easier, we cannot nullify the involved complexity and some of it is reflected in the developed components and their interplay.

**Improving the sorting.** There are also possibilities for further improvements of the visualization, in particular with regard to the sorting of rows according to segment similarity. Currently, the similarity is based on the edit distance, which basically counts the number of time steps with unequal labels. While this can be done quickly, it does not work well for short segments, which naturally have much less overlap. It is therefore advisable to consider alternative distance functions based on time-shifting [RLRS15].

Moreover, the TSP is currently solved with a nearest-neighbourhood heuristic. Although this is fast, it does not necessarily provide optimal solutions and might result in gaps or breaks in the vertical patterns. To address this issue, future work could study more closely the fact that the TSP to be solved is symmetric and metric. Alternative solution strategies tailored to this specific class of TSPs might lead to improved sorting results, while maintaining short calculation times [CGP12].

The sorting could also be improved in other views. For example, the axes in the PCSV, the rows in the TCDV or the cells in the TSCV can all be subject to interactive or automatic sorting based on different criteria, such as the importance of labels or the correlation values.

**Generalization to different application contexts.** It is worth mentioning that the developed approach is not limited to the dataset studied in this paper. There are no dataset-specific design decisions. In fact, our approach can be applied to any dataset that matches the introduced formalism around parameterizations, segmentations, labels and feature descriptors. Certainly, the concrete features might vary depending on the application context.

For example, as indicated by the expert feedback, it would be interesting to apply our approach to support the analysis of algorithms for the recognition of driver behaviour in the automotive sector. This would require new kinds of features and also the consideration of temporal subranges to concentrate the analysis more strongly on smaller changes in the driver behaviour.

**Considering uncertainty.** A final point for future work is to consider whether a parameter setting produces good or bad segmentations. To this end, it is necessary to expand the set of features by notions of segmentation accuracy [BBGM18]. This can involve features capturing label uncertainty or the overall number of errors in a segmentation.

9. Conclusion

In this paper, we presented a novel approach to visually understanding parameter dependencies in the context of time-series segmentation. The idea is to estimate which subranges of which parameters have an influence on which features of the segmented time series by calculating correlations. The correlations are then utilized to drive the interactive visual analysis. We introduced new visualizations that help analysts to understand not only if, but also how and in which subranges parameters exert influence on the segmentation. Corresponding interaction facilities support the interactive exploration and comparison of individual parameter dependencies. The application of our approach to activity recognition showed that the correlation-driven visual analysis can replicate previous findings and also generate new insight, which is otherwise difficult to obtain. The evaluation has shown that the approach addresses typical problems in time-series segmentation analysis and can provide good support in investigating parameter influence.

References

[ARHG10] AmirKhanov A., Heinzl C., Reiter M., Gröller E.: Visual optimality and stability analysis of 3DCT scan positions. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1477–1486. https://doi.org/10.1109/TVCG.2010.214.

[BAAK*13] Beyer J., Al-Awami A., Kathuri N., Lichtman J. W., Pfister H., Hadwiger M.: Connectomeexplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2868–2877. https://doi.org/10.1109/TVCG.2013.142.

[BBB*18] Bernard J., Bors C., Bögl M., Eichner C., Gschwandtner T., Miksch S., Schumann H., Kohlhammer J.: Combining the automated segmentation and visual analysis of multivariate time series. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2018), Eurographics Association, pp. 49–54. https://doi.org/10.2312/eurova.20181112.

[BBGM18] Bögl M., Bors C., Gschwandtner T., Miksch S.: Categorizing uncertainties in the process of segmenting and labeling time series data. In *EuroVis 2018 - Posters* (2018), The Eurographics Association, pp. 45–47. https://doi.org/10.2312/euro181126.

[BBDB*16] Bernard J., Dobermann E., Bögl M., Röhlig M., Vögele A., Kohlhammer J.: Visual-interactive segmentation of multivariate time series. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2016), Eurographics Association, pp. 31–35. https://doi.org/10.2312/eurova.20161121.

[Bis06] Bishop C. M.: *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006.

[BM10] Brückner S., Möller T.: Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1468–1476. https://doi.org/10.1109/TVCG.2010.190.

[BMPM12] Booshahr M., Möller T., Peterman R. M., Münzner T.: Vismon: Facilitating analysis of trade-offs, uncertainty,
and sensitivity in fisheries management decision making. *Computer Graphics Forum* 31, 3pt3 (2012), 1235–1244. https://doi.org/10.1111/j.1467-8659.2012.03116.x.

[BPFG18] BERGER W., PRINGER H., FILZMOSER P., GRÖLLER E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum* 30, 3 (2011), 911–920. https://doi.org/10.1111/j.1467-8659.2011.01940.x.

[BSM+13] BERGNER S., SEIDMAIR M., MÖLLER T., ADILOYOUSEF S. N., SAAD A.: Paraglide: Interactive parameter space partitioning for computer simulations. *IEEE Transactions on Visualization and Computer Graphics* 19, 9 (2013), 1499–1512. https://doi.org/10.1109/TVCG.2013.61.

[CGP12] CHAUHAN C., GUPTA R., PATHAK K.: Survey of methods of solving TSP along with its implementation using dynamic programming approach. *International Journal of Computer Applications* 52, 4 (2012), 12–19. https://doi.org/10.5120/8189-1550.

[EST18] EICHLER C., SCHUMANN H., TOMINISKI C.: Supporting visual parameter analysis of time series segmentation with correlation calculations. In *EuroVis 2018 - Posters* (2018), pp. 37–39. https://doi.org/10.2312/europ.20181124.

[GWR09] GUO Z., WARD M. O., RUNDENSTEINER E. A.: Model space visualization for multivariate linear trend discovery. In *IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 75–82. https://doi.org/10.1109/VAST.2009.5333431.

[Ham94] HAMBY D. M.: A review of techniques for parameter sensitivity analysis of environmental models. *Environmental Monitoring and Assessment* 32, 2 (1994), 135–154. https://doi.org/10.1007/BF00547132.

[HB03] HARROWER M. A., BREWER C. A.: Colorbrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. https://doi.org/10.1177/00087040.3235002042.

[HJS06] HELTON J. C., JOHNSON J. D., SALLABERRY C. J., STORLIE C. B.: Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety* 91, 10 (2006), 1175–1209. https://doi.org/10.1016/j.ress.2005.11.017.

[KNY*14] KRÜGER F., NYOLT M., YORDANOVA K., HEIN A., KRISTE T.: Computational state space models for activity and intention recognition, a feasibility study. *PLoS One* 9, 11 (2014), 1–24. https://doi.org/10.1371/journal.pone.0109381.

[Kul06] KULPA Z.: A diagrammatic approach to investigate interval relations. *Journal of Visual Languages and Computing* 17, 5 (2006), 466–502. https://doi.org/10.1016/j.jvlc.2005.10.004.

[LRHS14] LUBOSCHIK M., RYBACKI S., HAACK F., SCHULZ H.-J.: Supporting the integrated visual analysis of input parameters and simulation trajectories. *Computers & Graphics* 39 (2014), 37–47. https://doi.org/10.1016/j.cag.2013.09.004.

[MP13] MÜHLBACHER T., PRINGER H.: A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 1962–1971. https://doi.org/10.1109/TVCG.2013.125.

[PBCR11] PRETORIUS A. J., BRAY M.-A., CARPENTER A. E., RUDDLE R. A.: Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2402–2411. https://doi.org/10.1109/TVCG.2011.253.

[PKNM06] PITT M. A., KIM W., NAVARRO D. J., MYUNG J. I.: Global model analysis by parameter space partitioning. *Psychological Review* 113, 1 (2006), 57–83. https://doi.org/10.1037/0033-295X.113.1.57.

[QDV*12] QIANG Y., DELAFONTAINE M., VERSCHELDE M., DE MAEYER P., DE WEGHE N. V.: Interactive analysis of time intervals in a two-dimensional space. *Information Visualization* 11, 4 (2012), 255–272. https://doi.org/10.1177/1473871612436775.

[RLK*15] RÖHLM G., LUBOSCHIK M., KRÜGER F., SCHUMANN H., BÖGL M., ALSALLAKH B., MIKOSCH S.: Supporting activity recognition by visual analytics. In *IEEE Conference on Visual Analytics Science and Technology* (2015), pp. 41–48. https://doi.org/10.1109/VAST.2015.7347629.

[RLRS15] RICHTER C., LUBOSCHIK M., RÖHLM G., SCHUMANN H.: Sequencing of Categorical Time Series. In *IEEE Conference on Visual Analytics Science and Technology* (2015), pp. 213–214. Poster. https://doi.org/10.1109/VAST.2015.7347684.

[SHB*14] SEIDMAIR M., HEINZL C., BRUCKNER S., PRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2161–2170. https://doi.org/10.1109/TVCG.2014.2346321.

[STV17] STEEGEN S., TUERLINCKX F., VANPAEMEL W.: Using parameter space partitioning to evaluate a model’s qualitative fit. *Psychonomic Bulletin & Review* 24, 2 (2017), 617–631. https://doi.org/10.3758/s13423-016-1123-5.

[TWSM*11] TORSNEY-WIE T., SAAD A., MÖLLER T., HEGE H., WEBER B., VERBAVATZ J., BERGER S.: Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1892–1901. https://doi.org/10.1109/TVCG.2011.248.

[USKD12] UNGER A., SCHULTE S., KLEMMANN V., DRANSC D.: A visual analysis concept for the validation of geoscientific simulation models. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2216–2225. https://doi.org/10.1109/TVCG.2012.190.

[XG08] XU C., GERTNER G. Z.: Uncertainty and sensitivity analysis for models with correlated parameters. *Reliability Engineering & System Safety* 93, 10 (2008), 1563–1573. https://doi.org/10.1016/j.ress.2007.06.003.