Mining Characteristic Multi-Scale Motifs in Sensor-Based Time Series

ABSTRACT

More and more, physical systems are being fitted with various kinds of sensors in order to monitor their behavior, health or intensity of use. The large quantities of time series data collected from these complex systems often exhibit two important characteristics: the data is a combination of various superimposed effects operating at different time scales, and each effect shows a fair degree of repetition. For example, when monitoring a highway bridge, there will be seasonal and daily fluctuations in the *strain gauge* readings due to weather influences, as well as effects in the order of seconds or minutes due to individual vehicles and traffic jams. Each of these effects can be described by a small collection of *motifs*: recurring temporal patterns in the data. We propose a method to discover characteristic motifs at multiple time scales taking into account systemic deformations and temporal warping. In this method, we use a combination of scale-space theory and the Minimum Description Length principle to weigh the representational benefit of recognizing candidate motifs against the increased model complexity this may incur. We are thus guaranteed to report relevant temporal patterns without the risk of over-fitting the data. Our method will be validated on a number of real-world datasets, including a large series of strain gauge measurements from a concrete highway bridge and physiologic data collected during a snowboarding session.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Time Series, Motif Discovery, Minimum Description Length

1. INTRODUCTION

This paper is concerned with the discovery of temporal patterns in large time series, produced from physical sensors. In all but the most trivial applications, such sensor data will reflect the complexity of the physical system under investigation, and will show a combination of multiple effects. Some of these effects will be of interest, and central to the sensoring system, but others, such as noise and environmental effects, will merely be a disturbance and a hindrance to the identification of the phenomena of interest. The complex physical systems we aim to investigate here often have two important characteristics: a) multiple phenomena are at play in the sensor signal, and they typically occur at different time scales, b) each phenomenon will involve recurring events that will show up in the signal as repeating segments of data, often deformed and warped. In this paper, we propose a method that elegantly combines these two characteristics in order to discover recurring events at multiple time scales.

As a motivating example, we consider a Structural Health Monitoring (SHM) project involving huge quantities of sensor data collected at a highway bridge, similar to the data described in [7, 15]. Such bridge data fits our topic well, as it is subject to a number of effects that both show recurring events (traffic, daily temperature cycles), as well as largely varying time scales. The vertical displacement of the bridge, measured through *strain gauges*, is of course dependent on individual vehicles passing the bridge, over a period of several seconds (several tens of measurements). On a medium scale, the strain signal will show traffic jams, lasting up to an hour, that appear as clearly delineated intervals where the strain is increased due to the higher number of vehicles on the bridge. Finally, on a large scale, the strain is highly sensitive to the temperature of the bridge, such that the signal is dominated by a slow movement of the baseline, most notably with a day/night rhythm. Figure 1 shows 12 days of data collected at this bridge (some 10 million readings). Note that these different effects appear in a mixed fashion, and events at different time scales will often overlap. For example, traffic jams and individual vehicles will simply appear superimposed on the continually changing baseline of temperature effects on the strain. Additionally, vehicle peaks (shown as a detail on the right) will appear in the signal, even during traffic jams, as these often only affect one direction of traffic.

The recognition of repeating phenomena in time series is an important task in many applications, as it enables further processing of the data at a more conceptual level. For example, in SHM, it allows to determine traffic load statistics or various load-induced vibration patterns because it is vital

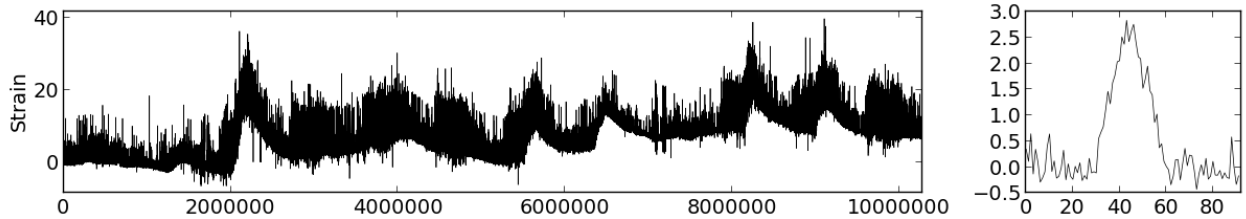

Figure 1: The plot on the left shows twelve days of strain measurements, sampled at 10 Hz from a highway bridge. The data exhibits recurring events, often superimposed, at multiple time scales, such as individual vehicles, traffic jams and daily fluctuations due to temperature changes. On the right it is shown an example of motif due to a passing vehicle.

to know when exactly certain events occur (such as heavy trucks). We assume that the recurring events will appear in a relatively small set of classes (e.g. trucks, cars), which we will refer to as *motifs*. The (scale-aware) motif discovery method presented here will then determine what the relevant motifs are, and when the different instances of each motif occur. In the specification of motifs, we intend to allow for a certain degree of flexibility in terms of duration and magnitude of the event. For example, a truck will be recognized as such, despite minor variations in speed and weight of the truck. Note that our definition of 'motif' is somewhat different from the use in other papers dealing with a similar problem [11], where a more strict matching based on Euclidean distance of a segment of fixed duration is employed. Although this approach works well in many scenarios, more flexibility is needed in the applications we consider.

In the motif discovery task in complex data, an important challenge we deal with is the possibility of superimposed events. Instances of motifs in one scale will overlap those in other scales, and the recognition of similar instances will be disturbed, if the possibility of multi-scale interference is not taken into account.

In this work, we propose an approach based on scale-space images [17] and the *minimum description length* (MDL) principle to address this problem [3]. The reason for choosing an MDL-based approach is that it allows us to find sets of motifs that represent a good trade-off between representation power and model simplicity. This guarantees that the reported motifs are actual recurring phenomena, rather than accidental coincidences, and that the motifs found are not too similar to each other. The novelty of our code, compared to a earlier approach [13] to the same problem, is that it explicitly supports the discovery of, potentially overlapping, multi-scale motifs.

The main contribution of our work is an algorithm for effectively finding multi-scale motifs that score well with respect to the MDL principle. Our algorithm combines several key ideas to achieve this:

- it uses *scale-space images* to characterize the contribution of the motifs at different temporal scales;
- it uses the *zero-crossings of derivatives* of the time series at different scales to identify repeating linear segments in the time series;
- it uses a *a symbolic representation* in combination with suffix trees to identify promising motifs consisting of these linear segments;
- it uses a greedy algorithm to select characteristic motifs that score well with respect to an MDL score.

We evaluate our method on a number of sensor-based time series from various applications. Results show that our approach can effectively discover a small set of characteristic motifs in the data, often directly related to particular events in the corresponding application domain.

The structure of this paper is as follows. Section 2 will introduce the notation and present necessary background information, including MDL, and the problem statement. In Section 3, we will motivate and define our method. Section 4 will evaluate and discuss experimental results. Section 5 presents related work. Finally, in Section 6, we draw conclusions and present ideas for future work.

2. BACKGROUND AND PROBLEM SETTING

In this section we introduce the notation, provide necessary background information and formally define the problem.

2.1 Notation and Preliminaries

We deal with finite sequences of numerical measurements (samples), collected by observing some property of a system with a sensor and represented as time series as defined below.

Definition 1. A **time series** of length n is an ordered sequence of values $\mathbf{x} = x[1], \ldots, x[n]$ of finite precision. A subsequence $\mathbf{x}[a:b]$ of $\mathbf{x}$ is defined as follows:

$$\mathbf{x}[a:b] = (\mathbf{x}[a], \mathbf{x}[a+1], \ldots, \mathbf{x}[b]),\ 1 \leq a < b \leq n$$

Moreover, and without loss of generality, we assume that the values are collected at a constant rate and none of them are missing and that the data has been z-normalized.

As motivated in the introduction, our goal is to find characteristic motifs in the input time series at multiple temporal scales. There are two equivalent ways of looking at motifs.

The first is that a motif is a structure that approximately repeats itself in a large number of places in the time series. The second is that a motif is a set of subsequences in the data, each pair of which is similar to each other [11]. We will refer to a structure that is approximately repeated in the data as a *motif*; subsequences of the data in which this motif occurs are referred to as *motif instances*.

An important feature of the motifs that we are looking for is that their instances can be warped or deformed to deal with potential slight variations in the duration and intensity of the events. This motivates our choice to represent motifs using linear segments as follows.

Definition 2. A **motif m** is a sequence of linear segments $[(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)]$, where a_i indicates the length of a segment (the duration) and b_i indicates the difference in value between the begin and end points of the segment.

In principle, higher order polynomials or other more complex functional representations may also be used to represent the segments, but we found that linear segments are simpler, have the advantage of avoiding overfitting, and are accurate enough in most cases.

We will be looking for instances of these motifs in the data.

Definition 3. Given a set of motifs M, let I be a function that maps a motif $\mathbf{m} \in M$ and a segment t of this motif to a set of subsequences of $\mathbf{x}$:

$$I(\mathbf{m}, t) = \{\mathbf{x}[a_{1t} : b_{1t}], ..., \mathbf{x}[a_{kt} : b_{kt}]\},$$

for some $a_{it}, b_{it} \in \{1, \ldots, n\}$ such that $a_{it} = b_{i(t-1)} + 1$ for $t > 1$ (i.e., the end of a segment determines the start of the next segment). Then $I(\mathbf{m}, t)$ determines the set of **instances** in $\mathbf{x}$ of segment t of motif $\mathbf{m}$.

Some choices for I are better than others; ideally instances closely resemble their associated motifs. The MDL score introduced in the next section will be used to evaluate the quality of a set of motifs M and of a function I.

Note that subsequences for the same motif and motif segment can have different lengths. This is necessary to deal with time warping.

From a high-level perspective, the problem that we are interested in is to identify a set of motifs that characterizes the data well. Taking into account the multi-scale nature of the data, it is desirable that instances of different motifs can overlap. In this way, one motif can reflect a regularity at a coarse scale, and another can reflect a regularity at a finer scale superimposed on top of the coarse structure. The next section defines more precisely how we evaluate a set of motifs and its instances to reflect these requirements.

2.2 Minimum Description Length

Our main idea is to approach the problem of selecting motifs as a *model selection* problem. This allows us to employ the Minimum Description Length [3] principle to rank motifs. MDL is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data. In our setting, a model consists of a set of motifs M. Following the two-part MDL principle, the best set of motifs to describe the time series $\mathbf{x}$ is the one that minimizes the sum $L(M) + L(\mathbf{x} \mid M)$, where

- $L(M)$ is the length, in bits, of the description of the motifs, corresponding to a model;
- $L(\mathbf{x} \mid M)$ is the length, in bits, of the description of the time series when encoded with the help of the motifs M, that is the residual information not represented by M.

In order to apply the MDL principle in practice, we need to define an encoding scheme for a given set of motifs M and, consequently, how to compute both $L(M)$ and $L(\mathbf{x} \mid M)$. However, we need first to clarify how we discretize the time series as the MDL principle is only applicable to discrete data. Both aspects are addressed in the following sections.

2.2.1 Time Series Values Discretization

In order to use the MDL principle, we need to work with a quantized input signal. Because of this, we assume that the values $\mathbf{x}[i]$ of the input time series $\mathbf{x}$ have been quantized to a finite number of symbols by employing the function defined below:

$$Q(\mathbf{x}[i]) = \left\lfloor \frac{v - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}\, l \right\rfloor - \frac{l}{2}$$

where l, assumed to be even, is the number of bins to use in the discretization while $\min(\mathbf{x})$ and $\max(\mathbf{x})$ are respectively the minimum an maximum value in $\mathbf{x}$. Throughout the rest of the paper, we assume $l = 256$.

One question that might arise is if such a quantization removes meaningful information from the time series. In [4] the authors show that the effect of quantization is rather modest on several time series from various domains.

2.2.2 Encoding of the Model

We will first discuss the encoding of the model, i.e. a set of motifs M. Each motif essentially consists of a sequence of linear segments, each described by two integers. The length of a segment cannot be longer than the total length of the time series; hence, we use $\log_2 n$ bits to encode it. The difference in value between the begin and end point is limited by the quantization used; in our setting 8 bits are sufficient. Finally, with $\log_2 n$ bits we can encode the number of segments in a motif. Summing up we have

$$L(M) = \sum_{i=1}^{m} (\log_2 n + k_i(\log_2 n + 8)),$$

where m is the number of motifs and k_i is the number of segments in motif i. We assume that these motifs are ordered in the encoding. We use this order to distinguish the scales at which the motifs are present.

2.2.3 Encoding the Data

We will now describe how we compute $L(\mathbf{x} \mid M)$, that is the description length of the time series when encoded with the help of a set of motifs M. In the definition of the code we will also use the instances I associated to each motif in M. Our assumption is that a good selection of motifs M and associated instances I will help to encode the data more concisely.

We will first define the entropy of a time series as it is a key concept we will need in the following paragraphs.

Definition 4. The **entropy** of a time series $\mathbf{x}$, discretized according to a set of values D, is defined as below

$$H(\mathbf{x}) = -\sum_{v \in D} P(\mathbf{x}[i] = v) \log_2 P(\mathbf{x}[i] = v)$$

where $P \log_2 P = 0$ in the case of $P = 0$ and $P(\mathbf{x}[i] = v)$ indicates the fraction of points in the time series which has value v.

Given the definition of entropy, we can define the description length of a time series as follows, assuming we have not identified any motifs.

Definition 5. Given a time series $\mathbf{x}$ of length n, its **description length** (in bits) is given by

$$L(\mathbf{x}) = nH(\mathbf{x})\,.$$

Our main idea is now that a good choice of motifs M and associated instances $I(\mathbf{m}, t)$ should lead to a code for the time series with a description length shorter than $L(\mathbf{x})$. To this aim, we introduce a code for $\mathbf{x}$ given a choice of M and $I(\mathbf{m}, t)$. It will be the task of the search algorithm to determine the best configuration.

Concretely, for each chosen motif $\mathbf{m}$ and corresponding motif instances $I(\mathbf{m}, t)$, we first encode the time stamps and the (vertical) values at which the instances of the first segment of $\mathbf{m}$ start. For one motif with ℓ instances, this requires $\log_2 n + \ell(\log_2 n + 8)$ bits, where $\log_2 n$ bit are needed to encode the number of instances, and $\log_2 n$ and 8 are the bits needed to code the starting time stamp and vertical value, respectively.

There are instances for each segment in a motif. While encoding these, we need to allow for a certain amount of time warping, and hence the segment in each instance may deviate both in length and in amplitude from the segment in the motif. Both vertical and horizontal differences from the segment in the motif can be represented by sequences of integers: the deviations of segment lengths can be represented in one sequence $[a_{ijk} \mid 1 \leq i \leq m, 1 \leq j \leq \ell_i, 1 \leq k \leq k_i]$, where m is the number of motifs, ℓ_i the number of instances of motif i and k_i the number of segments in the motif; similarly, the differences in value can be listed. In order to favor only small numbers of values, we compute the description length of these sequences employing an entropy-based encoding as in Definition 5.

This code for motifs and instances leads to an approximation of the data, as follows. For each position in the time series, we determine the last motif in the ordered set of motifs which has an instance at this position. Whether a position is covered by an instance is determined by taking into account the starting positions of the first segment of the motif, the lengths of the other segments in the motifs, and the deviations from these lengths as encoded in the code of deviations. The reason for using the order of motifs is that we explicitly allow motifs to overlap. This allows us to deal with the multi-scale aspects of the data.

The approximated value of a position in the time series covered by a motif is determined by linear interpolation between the two end points of the motif segment in which the position is included. These end points are determined similarly from the encodings of locations, motifs and deviations.

Our remaining code for the data now consists of two parts. First, for each position in the data covered by a motif, the error is encoded with respect to the approximation. An entropy encoding is used for these errors. Second, for the remaining time stamps, which are not covered by a motif, an entropy encoding is used as well to code the original value for that position.

Note that in this code we have a constant number of dictionaries (for duration, difference in value, errors, and remaining original time points). Hence, we do not need to calculate the size of these dictionaries explicitly.

The final description length $L(\mathbf{x} \mid M)$ is given by the sum of the lengths (in bits) of the code components described above.

2.3 Problem Statement

We have now introduced the necessary definitions and background material to state our problem.

> Given a time series $\mathbf{x}$, we want to find a set of motifs M and associated instances, such that the sum $L(M) + L(\mathbf{x} \mid M)$ is minimized.

Clearly, this problem is hard to solve exactly. Hence, in the next section we define a step-wise heuristic algorithm that works well in practice.

3. MOTIF SELECTION ALGORITHM

The proposed heuristic motif discovery algorithm consists of several steps, which will be shown to perform well in the next section. The first steps will identify a set of promising candidate motifs; the last steps select a characteristic subset of the motifs based on the MDL scoring function discussed earlier. Figure 2 shows a high level overview of our method and the steps involved.

3.1 Finding Candidates Motifs

In this section we describe our candidate motif generation procedure. Several key ideas underly this procedure.

- It uses the *scale-space image* to characterize the contribution of the motifs at different temporal scales;

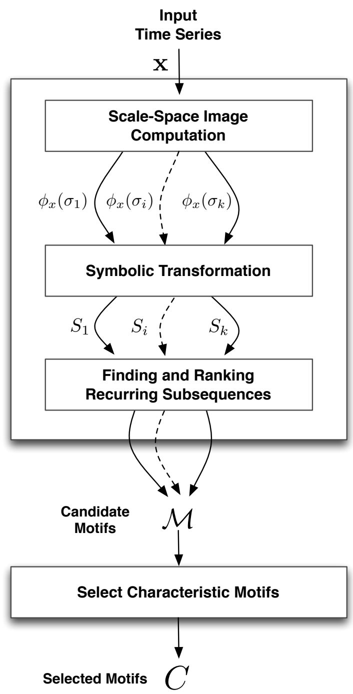

Figure 2: Overview of the proposed motif discovery and selection algorithm.

- It effectively identifies promising segments at multiple scales by discretizing the time series using *the derivatives of the signal in scale-space* in combination with *k-means clustering*;
- In the discretized representation, it merges recurring sequences of adjacent segments by employing a *suffix tree* based approach.

The subsequent sections discuss this in more detail.

3.1.1 Scale-Space Image

Scale-space images [17] are a widely used scale parameterization technique for one-dimensional signals[1] based on the operation of Gaussian convolution. We use them to characterize the contribution of the motifs at increasingly higher temporal scales while, at the same time, removing (smoothing out) the effect of the motifs at finer scales. We start by giving the definition of convolution as presented in the signal processing literature.

Definition 6. Given a signal $\mathbf{x}$ of length n and a response function (kernel) $\mathbf{h}$ of length m, the result of the **convolution $\mathbf{x} * \mathbf{h}$** is the signal $\mathbf{y}$ of length n, defined as:

$$\mathbf{y}[t] = \sum_{j=-m/2+1}^{m/2} \mathbf{x}[t-j]\,\mathbf{h}[j]$$

[1]From now on, we will use the term signal and time series interchangeably.

When referring to Gaussian convolution, $\mathbf{h}$ is defined as the Gaussian kernel having mean $\mu = 0$, standard deviation σ and area under the curve equal to 1, discretized into m values.[2] Moreover, $\mathbf{x}$ is padded with $m/2$ zeros before and after its defined range to account for boundary effects, as $\mathbf{x}[t-j]$ may be undefined for some j. Lindeberg [10] provides a detailed review on how to compute Gaussian convolutions for discrete signals.

The Gaussian convolution essentially smooths each value $\mathbf{x}[t]$ according to its neighboring values. The amount of removed detail is directly proportional to the standard deviation σ (and thus the kernel size m), from now on referred to as the *scale parameter*. In the limit, when $\sigma \to \infty$, the result of the Gaussian convolution converges to the mean of the signal $\mathbf{x}$. We can now formally introduce the scale-space image.

Definition 7. Given a signal $\mathbf{x}$, its **scale-space image** is the family of σ-smoothed signals $\Phi_{\mathbf{x}}$ over the scale parameter σ defined as follows:

$$\Phi_{\mathbf{x}}(\sigma) = \mathbf{x} * \mathbf{g}_\sigma\,,\ \sigma > 0$$

where $\mathbf{g}_\sigma$ is a Gaussian kernel having standard deviation σ, and $\Phi_{\mathbf{x}}(0) = \mathbf{x}$.

We quantize the scale-space image across the scale dimension by computing the Gaussian convolutions only for a finite number of scale parameters. More formally, for a given signal $\mathbf{x}$, we consider a set of scale parameters S and we compute $\Phi_{\mathbf{x}}(\sigma)$ only for $\sigma \in S$. The number of scale parameters considered, and thus the resolution of the quantization, depends on the final application and on the distribution of the motifs across the scale dimension. If, for instance, the motifs appear at considerably different scales, a coarser quantization would suffice to isolate them across the scale dimension. On the other hand, if different motifs appear at similar scales, a finer quantization is needed to effectively separate their corresponding contributions to the signal. In order to support both scenarios, we define two sets of scale parameters $S_{coarse} = \{2^i \mid 0 \le i \le \sigma_{max} \ \wedge i \in \mathbb{N}\}$ and $S_{fine} = \{\sqrt{2^i} \mid 0 \le i \le 2\sigma_{max} \ \wedge i \in \mathbb{N}\}$ which well adapt to the practical cases we consider.

As an example, Figure 3 shows the scale-space image computed from an artificially generated signal with $S = S_{coarse}$. The topmost plot represents the original signal, constructed by three components at different temporal scales: a slowly changing and slightly curved baseline, medium term motifs (bumps) and short term motifs (peaks). By visual inspection, it can be easily verified that, by increasing the scale parameter, a larger amount of detail is removed. In particular, the peaks disappear at scales greater than $\sigma = 2^4$ while the bumps are smoothed out at scales greater than $\sigma = 2^8$, after which only the baseline contributes to the scale component.

We now deal with multi-scale aspect of the data by identifying motifs in each of the scales in the scale image.

[2]To capture almost all non-zero values, we define $m = \lfloor 6\sigma \rfloor$.

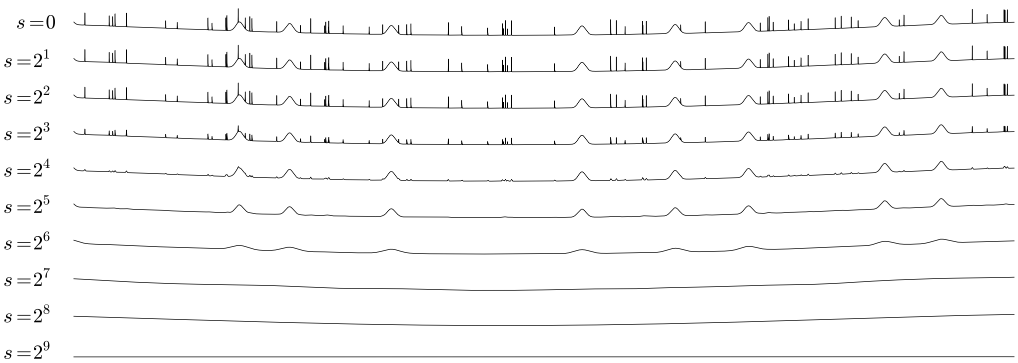

Figure 3: Scale-space image of an artificially generated signal consisting of two motifs at different temporal scales and a slowly changing baseline.

3.1.2 Finding Candidate Segments

Before identifying candidate motifs, we first identify candidate linear segments. A useful tool to quickly identify promising boundaries for linear segments in the time series are the *zero-crossings of derivatives.*

Given a time series $\mathbf{x}$ and one of the components of its scale-space image $\Phi_{\mathbf{x}}(\sigma)$, let

$$z(j) = \{t_1, \ldots, t_m\}, \text{such that } \frac{d^j \Phi_{\mathbf{x}}(\sigma)}{dt}(t_i) = 0,$$

$$Z = z(1) \cup \cdots \cup z(d_{max})$$

be the sorted locations in $\Phi_{\mathbf{x}}(\sigma)$ of the zero-crossing of its derivatives until order d_{max}. Note that d_{max} will typically be low, e.g. just 1 or 2.

These zero-crossings are informative as they indicate points in the time series at which the direction of the signal changes; these positions are good candidates for a change of the linear coefficients as well. Thus, each segment bounded by two consecutive zero-crossings could be an instance of a segment in a motif. We use $k-$means clustering to identify a small set of prototype segments, as follows. Each segment between zero-crossings can be thought of as a data point in a feature space, where the features are the duration and difference in value between the zero-crossings of the derivatives. More precisely, we consider the data points $F_{\Phi_{\mathbf{x}}(\sigma)} = \{\mathbf{f}_i = (h_i, v_i)\}$ where

$$h_i = t_{i+1} - t_i \, , \; 1 \leq i < n$$

is the time between each pair of consecutive zero-crossings and

$$v_i = \Phi_{\mathbf{x}}(\sigma)[t_{i+1}] - \Phi_{\mathbf{x}}(\sigma)[t_i] \, , \; 1 \leq i < n$$

is their vertical distance. Figure 4 illustrates this concept.

These data points are clustered using the $k-$means clustering algorithm, where k is a parameter that determines the number of candidate segments. Preliminary experiments show that setting the parameter k in practice is not a critical problem.

The centers of the identified clusters are the candidate reference segments, which will be combined into motifs in the next step. Note that the clustering algorithm ensures that candidate segments will be not too dissimilar from each other. This procedure is repeated for each scale in the scale-space independently.

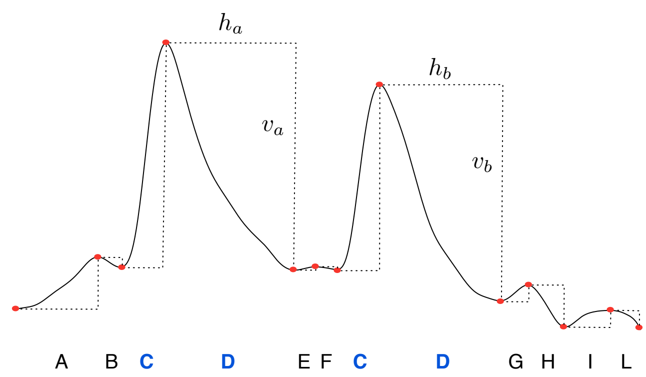

Figure 4: Example of the feature space based on the zero-crossings of the derivatives (only order one in this figure) and the clustered candidate segments identified by letters.

3.1.3 Finding Candidate Motifs

The key idea in identifying motifs is to represent time series symbolically (see Figure 4). Each symbol in this representation corresponds to the candidate segment identified by the $k-$means algorithm for that segment.

After transforming each scale-space image component into the symbolic representation defined above, we identify motifs by looking for repeating subsequences in the obtained string as similarly done by previous approaches [6, 8], although using different kinds of representations such as SAX [9].

Our candidate motifs generation procedure is summarized in pseudo-code in Algorithm 1. `ScaleSpaceImage`$(\mathbf{x}, S)$ returns the scale-space image of $\mathbf{x}$ defined over the scale parameters S. `ComputeZeroCrossings`$(\Phi_{\mathbf{x}}(\sigma_i), d_{max})$ calculates the zero-crossings of the derivatives for each scale. `SymbolicQuantization`$(\Phi_{\mathbf{x}}(\sigma_i), Z, A)$ transforms each time

Algorithm 1 Find candidate motifs

Input: a time series $\mathbf{x}$, a set of scales parameters $S = \{\sigma_1, ..., \sigma_k\}$, the maximum order for the derivatives roots d_{max}, the cardinality A of the symbolic representation, the number of motifs considered per scale r

Output: a set of candidate motifs $\mathcal{M} = \{M_{s,r}\}$ indexed by scale parameter s and rank r.

$\mathcal{M} = \{\}$
$\Phi_{\mathbf{x}}(\sigma_1), \ldots, \Phi_{\mathbf{x}}(\sigma_k) = \texttt{ScaleSpaceImage}(\mathbf{x}, S)$
for $i = 1 \ldots k$ **do**
  $Z_i = \texttt{ComputeZeroCrossings}(\Phi_{\mathbf{x}}(\sigma_i), d_{max})$
  $S_i = \texttt{SymbolicQuantization}(\Phi_{\mathbf{x}}(\sigma_i), Z_i, A)$
  $\Sigma_i = \texttt{FindRecurringSubstrings}(S_i)$
  $M_{\sigma_i,r_1}, ..., M_{\sigma_i,r_m} = \texttt{RankMotifsByCoverage}(\Sigma_i, r)$
  $\mathcal{M} = \mathcal{M} \cup \{M_{\sigma_i,r_1}, ..., M_{\sigma_i,r_m}\}$
end for

series $\Phi_{\mathbf{x}}(\sigma_i)$ into a symbolic string given the zero-crossings Z and cardinality A. $\texttt{FindRecurringSubstrings}(S_i)$ returns the set of all maximal substrings of length at least 2 that appear at least twice in the data (maximal in the sense that no longer substring occurs twice). In general, we could parameterize this; however, in our experiments we found these parameters to work in all cases. Furthermore, an important advantage of this setup is that we can calculate this set of substrings in linear time by using suffix trees. $\texttt{RankMotifsByCoverage}(\Sigma_i, r)$ selects the best scoring r motifs from this set of substrings. The evaluation is as follows: the occurrences of each string in the time series are determined; these occurrences are mapped back to the original time series; the total length of the original time series covered by these occurrences is determined. The main motivation is that we can expect the best coding motifs to be those that cover large parts of the time series. The final selection from the resulting set of candidate motifs is done in the next step.

3.2 Selecting Characteristic motifs

The naive way to select the best set of motifs would be to enumerate all potential subsets and choose the one that minimizes the sum $L(M) + L(\mathbf{x} \mid M)$. However, the space of motif sets grows exponentially with the number of candidate motifs and this makes an exhaustive evaluation computationally infeasible for large time series. Because of this, we propose a heuristic selection strategy that overcomes these computational limitations. Our motif selection heuristic is shown in pseudo-code in Algorithm 2.

Algorithm 2 Select characteristic motifs

Input: a time series $\mathbf{x}$, a set of candidate motifs $\mathcal{M} = \{M_{s,r}\}$ indexed by scale parameter s and rank r.

Output: a set of selected motifs $C \subseteq \mathcal{M}$.

$C = \{\}$
for $i = k \ldots 1$ **do**

$$j = \arg \min_{j \in \{1,\ldots,m\}} L(C \cup \{M_{\sigma_i,j}\}) + L(\mathbf{x} \mid C \cup \{M_{\sigma_i,j}\})$$

  $C = C \cup \{M_{\sigma_i,j}\}$
end for

Essentially this algorithm traverses the candidate motifs starting at the coarsest scale and, for each scale, it adds the motif that improves the MDL score the most.

3.3 Computational Complexity

The construction of the scale-space image requires to compute $|S|$ convolutions. This can be done efficiently using the Fast Fourier Transform in $O(|S|\, n \log_2 n)$ time. The computation of the zero-crossing of the derivatizes can be done with a linear scan and thus has $O(n)$ complexity. The complexity of the symbolic transformation, carried out by k-means in $O(Ik\,|Z|)$ time depends on the number of zero-crossings features to cluster which, given a property of the scale-space image [17], can only decrease as the scale is increased; here I is the number of iterations of the $k-$means algorithm. Preliminary experiments even show that the decrease in $|Z_i|$ is exponential. Locating recurring substrings in the symbolic representation can be done in linear time employing a suffix tree; the number of such strings ($|\mathcal{M}|$) is $O(n)$ in the worst case and much smaller in practice. We calculate the instances of the corresponding motifs in $O(n)$ time for each motif identified. Sorting the resulting motifs takes $O(|\mathcal{M}| \log |\mathcal{M}|)$ time. During the final traversal of this set, we need to calculate the MDL score for each intermediate model. This calculation takes $O(|C|n)$ time; note that the size of the dictionaries can be considered constant. Overall, this gives our method a complexity of $O(n \log_2 n + |\mathcal{M}|(\log |\mathcal{M}| + |C|n)$ time.

4. EXPERIMENTAL EVALUATION

In this section, we evaluate our method experimentally, on two real-life sensor datasets, one describing physical exercise, and one collected from the sensor network of the highway bridge mentioned in the introduction. Moreover, we compare our method with another published approach on a common dataset.

4.1 Snowboard Data

The first experiment relates to physiologic data collected during a day of snowboarding in the Austrian Alps. The data was collected by a Zephyr BioHarness[3] 3 breast strap, which monitors several key physiological parameters and logs them at a sampling rate of 1 Hz. Alpine sports are an interesting domain for our method, as it naturally contains the cyclic phenomenon of ascending by ski lift and descending 'on foot'. This produces a recurring pattern of intense exercise while descending and clear signs of recuperation while being transported up. Especially when the same lift and slope are repeatedly taken, this will lead to motifs in the measured time series. Additionally, on a smaller scale, the natural tendency of the human body is to introduce shorter cycles of activity and rest, especially when dealing with intense activity and high altitude.

The data considered here describes heart rate measurements taken during 2.5 hours of mixed activity, starting at 11:00 AM, with some 40 minutes actually spent on the slopes. We employed S_{fine} as scale parameters, set $d_{max} = 1$ and the cardinality of the symbolic representation to 10. Figure 5 shows two key selected motifs, which correspond to the phenomena described above. The top motif represents some 16 minutes, corresponding to recuperation (decreasing heart rate while on the lift), exercise and recuperation again. A full cycle of ascent and descent takes about 10

[3]http://www.zephyranywhere.com/products/bioharness-3

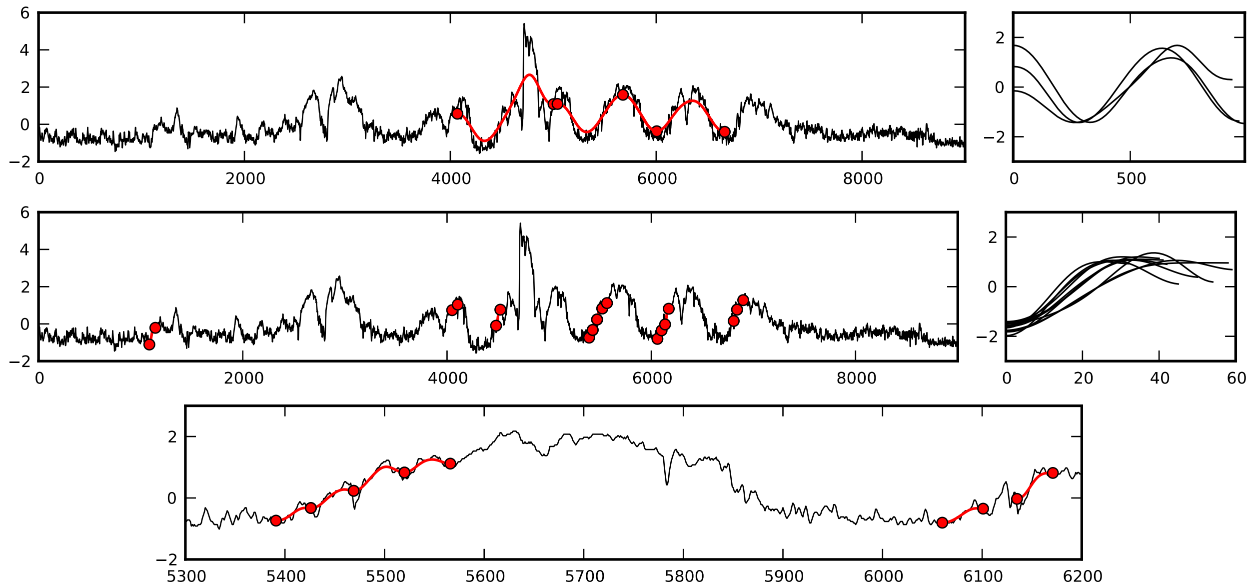

Figure 5: Selected motifs in the Snowboard data. Left side: motif occurrences in the series. Right side: motifs at the respective scale-space component after z-normalization.

minutes, which corresponds with the manual annotations. This pattern occurs three times in this dataset, at the scale component $\Phi_x(\sqrt{2^{14}})$, as indicated by the red segments in the diagram. Note that two instances actually overlap, as the motif describes more than a single cycle. These two instances actually relate to two descents of a single slope. The second motif, at the scale component $\Phi_x(\sqrt{2^7})$, has 10 instances of increasing and then decreasing heart rate, presumably related to short exercise intervals of around 50 sec. A detail of this motif is shown in the bottom diagram, showing just 20 minutes at 12:25.

The overall number of scale components considered for this data is 22 for a total of 13 selected motifs. However, motifs selected at scales greater than 2^{16} did not show motifs relevant to this particular application domain.

4.2 Highway Bridge Data

We subsequently evaluate our approach on the time series data previously shown in Figure 1. The series has been collected in the context of a Structural Health Monitoring project,and consists of 12 days of strain measurements (for a total of $10,280,939$ data points) from one span of the monitored highway bridge. As the bridge is affected by several phenomena operating at multiple time scales, the strain measurements contain various classes of recurring motifs reflecting this fact and represents an ideal dataset to test our method. We employed S_{coarse} as scale parameters and set $d_{max} = 1$ and the cardinality of the symbolic representation to 10. Figure 6 shows two of the most interesting selected motifs, respectively at scale components $\Phi_x(2^3)$ and $\Phi_x(2^{15})$. The first motif identifies the most recurring events in the data, i.e. passing vehicles. In the graph, a red pixel is drawn for each instance, for a total of $58,646$ occurrences, which cover almost 22% of the data. On the right, we plot all the motif instances (after normalization) superimposed, as represented in the scale component $\Phi_x(2^3)$. The selected motif represents a high variability of instances, in both duration and amplitude, that can be directly related to the speed and weight of the vehicles. This information can thus be used by bridge managers to evaluate the load patterns of the infrastructure and potentially aid the decision making when planning maintenance activities. The second motif represents a much longer pattern occurring on a daily basis due to changes in temperature that, in turn, affect the response of the bridge to external forces. A total of 5 motif instances of this kind occur, covering around 24% of the data. Note how occurrences of the first motif are superimposed over the instances of this one.

The overall number of scale components considered is 19, although the motifs selected at scales greater than 2^{17} are not of any interest in relation to the application domain.

4.3 Comparison with related work

To the best of our knowledge, there are no published methods dealing with the discovery of characteristic sets of multi-scale and overlapping motifs in time series data. As we cannot compare our method with others in a multi-scale setting, we chose to also evaluate our algorithm on a time series presented in [13], in which no multi-scale events are present. A comparison on such data is of interest as our method should be able to identify the non-overlapping motifs present in this data as well.

The considered time series was produced by extracting the first MFCC coefficients from an audio file featuring two repeated kinds of bird calls, resulting in two motifs present in the data. The time series has a total of 1367 measurements. As the motifs in the data are rather similar in length, we do not need to consider the whole scale-space image. Instead, we set the scale parameters to $S = \{1, \sqrt{2}, 2, \sqrt{8}\}$. The result shown here was obtained by setting the cardinality of the symbolic representation to 6. However, in order

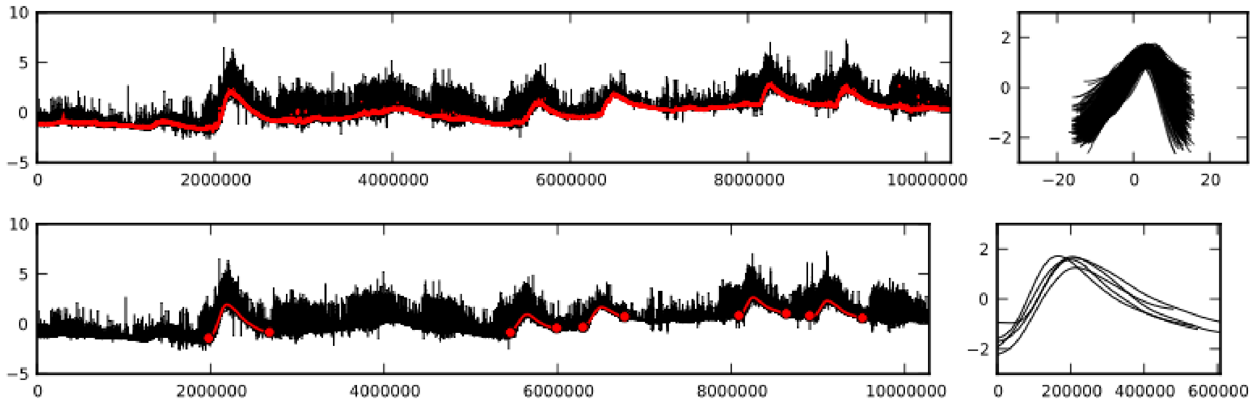

Figure 6: Selected motifs in the highway bridge data. Left side: motif occurrences in the series. Right side: motifs at the respective scale-space component after z-normalization.

to assess the sensitivity of the method in relation to the size of the alphabet, we tried cardinalities ranging from 5 to 15 obtaining qualitatively similar results. Figure 7 reports the motifs selected by our method. These motifs are similar to those obtained by the clustering method proposed in [13] for non-overlapping motifs. Although in this case we manually specified the scale parameters, we note that the algorithm in [13] also requires to provide an educated guess of parameters, i.e. of the approximate lengths of the motifs to look for.

5. RELATED WORK

The problem of discovering recurring temporal patterns in time series data is an important one and has received considerable attention by the community from different perspectives.

Subsequence Clustering. Early work considers the related problem of clustering the (overlapping) subsequences in the time series extracted through a sliding window. Subsequence clustering is an obvious and intuitive choice for finding characteristic subsequences in time series. However, this approach requires the a priori specification of the lengths of the subsequences to consider and is not generally tailored to support multi-scale data. Moreover, in a paper by Keogh et al. [5], it was shown that, despite the intuitive match, subsequence clustering is prone to a number of undesirable behaviors that makes the end result meaningless and independent of the data at hand. A number of papers [1, 2, 16] have further investigated the observed phenomena, providing solutions to overcome it. Yet, since the publication of [5], the subsequence clustering idea has seen a serious decline in popularity. In [5], the authors proposed a solution based on motif discovery.

Motif discovery and clustering. Motif discovery has received a fair amount of attention, in particular after subsequence clustering was shown to be unreliable. In [11], a motif is defined rather strictly as the pair of most similar subsequences in a time series according to the Euclidean distance, and the authors propose an efficient and exact method to find such pairs. Saria et al. [14], on the other hand, propose a more flexible definition of motif, based on a shape template that can be affected by non-linear transformations such as temporal warping and additive noise. They introduce an unsupervised algorithm to discover the set of canonical shape templates in the data. Although the method is able to discover motifs of different lengths, it does not deal with multi-scale data where multiple motifs at different time scales could appear superimposed.

To the best of our knowledge, the most similar work to ours is [13]. The authors propose a method to mine a set of clusters of motifs from a given time series. The clusters are formed according to an agglomerative procedure. First, a single cluster is created containing the pair of most similar subsequences in the data (this is done with repeated runs of the exact motif discovery algorithm introduced in [11]). After that, the set of clusters is iteratively refined by taking one of the following actions: create a new cluster, add to a cluster, merge two clusters. The algorithm looks for the best operator to apply such that the MDL score for the clusters set is lowered, or it stops otherwise. This method does not however consider superimposed motifs like those found in the multi-scale data we consider in this paper.

Multi-scale Time Series Data. Although several papers address the problem of discovering recurring patterns in time series, few of them consider data where combinations of effects at multiple temporal scales affect the patterns or motifs. In [12], Papadimitriou et al. propose a method to discover the key trends in a time series at multiple time scales (window lengths) by introducing an incremental version of Singular Value Decomposition. Vespier et al. [15] propose an MDL-based method to recognize the most relevant scales of analysis in the data and, consequently, to separate the time series into distinct components. This method does not however characterize the individual motifs directly, but rather assesses the relevancy on the informative content present at each temporal scale.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a method for the discovery of multi-scale recurring patterns (motifs) in time series data. Our work is motivated by a SHM project which deals with high-frequency measurements collected by a sensor network

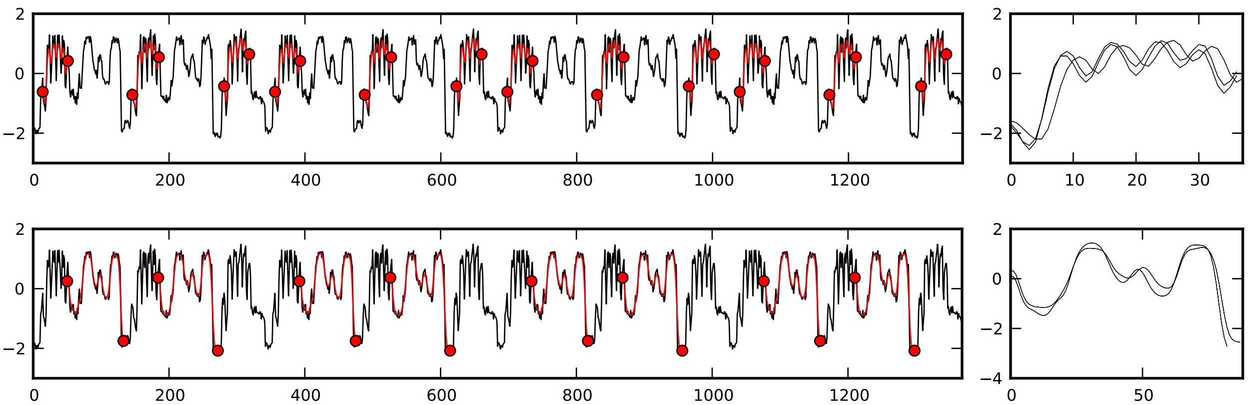

Figure 7: Selected motifs in the *bird calls* data from [13]. Left side: motif occurrences in the series. Right side: motifs at the respective scale-space component after z-normalization.

deployed on a highway bridge. In particular, we focused on a property that sensor data collected from complex systems typically exhibits: the presence of multiple phenomena at play in the sensor signal, often occurring at different time scales and potentially superimposed and mixed together. Because of the high degree of variability present in this kind of data, we have adopted a definition of motif based on structural complexity other than on point-wise similarity (i.e. Euclidean distance) as in much previous work. In order to discover the most characteristic recurring motifs, we proposed an algorithm based on a combination of scale-space theory, string processing and the Minimum Description Length principle. We showed the effectiveness of our method on sensor data from several applications.

Future work includes evaluating our method on additional data exhibiting multi-scale behavior, as a few datasets of this kind are currently publicly available. Moreover, we are interested in further developing the symbolic representation we adopted, currently requiring the cardinality of the alphabet as a parameter; ideally, our method would become parameter free.

7. REFERENCES

[1] J. R. Chen. Making clustering in delay-vector space meaningful. *Knowl. Inf. Syst.*, 11(3):369–385, Apr. 2007.

[2] A. M. Denton, C. A. Besemann, and D. H. Dorr. Pattern-based time-series subsequence clustering using radial distribution functions. *Knowl. Inf. Syst.*, 18(1):1–27, Jan. 2009.

[3] P. Grünwald. *The minimum description length principle*. The MIT Press, 2007.

[4] B. Hu, T. Rakthanmanon, and Y. Hao. Discovering the intrinsic cardinality and dimensionality of time series using mdl. In *Proceedings of ICDM '11*, 2011.

[5] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowl. Inf. Syst.*, 8(2):154–177, Aug. 2005.

[6] E. Keogh, S. Lonardi, and B. Y.-c. Chiu. Finding surprising patterns in a time series database in linear time and space. *Proceedings of KDD '02*, page 550, 2002.

[7] A. Knobbe, H. Blockeel, A. Koopman, T. Calders, B. Obladen, C. Bosma, H. Galenkamp, E. Koenders, and J. Kok. InfraWatch: Data management of large systems for monitoring infrastructural performance. *Advances in Intelligent Data Analysis IX*, pages 91–102, 2010.

[8] J. Lin, E. Keogh, and S. Lonardi. Visualizing and Discovering Non-Trivial Patterns In Large Time Series Databases. *Information visualization*, 2005.

[9] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, Oct. 2007.

[10] T. Lindeberg. Scale-space for discrete signals. *Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, Mar. 1990.

[11] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. In *Proceedings of SDM '09*, pages 473–484, 2009.

[12] S. Papadimitriou and P. Yu. Optimal multi-scale patterns in time series streams. In *Proceedings of SIGMOD '06*, 2006.

[13] T. Rakthanmanon, E. Keogh, S. Lonardi, and S. Evans. Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data. In *Proceedings of ICDM '11*, 2011.

[14] S. Saria, A. Duchi, and D. Koller. Discovering deformable motifs in continuous time series data. In *Proceedings of IJCAI '11*, 2011.

[15] U. Vespier, A. Knobbe, S. Nijssen, and J. Vanschoren. MDL-Based Analysis of Time Series at Multiple Time-Scales. In *Proceedings of ECML PKDD '12*, 2012.

[16] U. Vespier, A. Knobbe, and J. Vanschoren. Traffic events modeling for structural health monitoring. In *Proceedings of IDA '11*, 2011.

[17] A. Witkin. Scale-space filtering. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 329–332, 1987.