# MDL-Based Analysis of Time Series at Multiple Time-Scales

Ugo Vespier[1], Arno Knobbe[1], Siegfried Nijssen[2], and Joaquin Vanschoren[1]

[1] LIACS, Leiden University, The Netherlands
[2] Katholieke Universiteit Leuven, Belgium
uvespier@liacs.nl

**Abstract.** The behavior of many complex physical systems is affected by a variety of phenomena occurring at different temporal scales. Time series data produced by measuring properties of such systems often mirrors this fact by appearing as a composition of signals across different time scales. When the final goal of the analysis is to model the individual phenomena affecting a system, it is crucial to be able to recognize the right temporal scales and to separate the individual components of the data. In this paper, we approach this challenge through a combination of the Minimum Description Length (MDL) principle, feature selection strategies, and convolution techniques from the signal processing field. As a result, our algorithm produces a good decomposition of a given time series and, as a side effect, builds a compact representation of its identified components. Experiments demonstrate that our method manages to identify correctly both the number and the temporal scale of the components for real-world as well as artificial data and show the usefulness of our method as an exploratory tool for analyzing time series data.

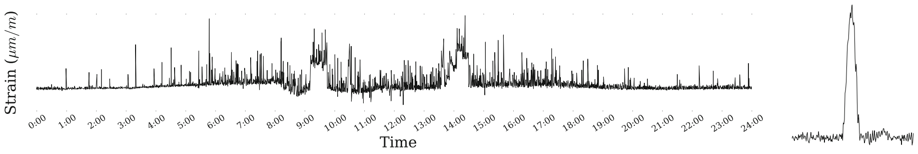
**Keywords:** Time Series, Scale Selection, Minimum Description Length.

## 1 Introduction

This paper is concerned with the analysis of sensor data. When monitoring complex physical systems over time, one often finds multiple phenomena in the data that work on different time scales. If one is interested in analyzing and modeling these individual phenomena, it is crucial to recognize these different scales and separate the data into its underlying components. Here, we present a method for extracting the time scales of various phenomena present in large time series. The method combines concepts from the signal processing domain with feature selection and the Minimum Description Length principle [2].

The need for analyzing time series data at multiple time scales is nicely demonstrated by a large monitoring project in the Netherlands, called *InfraWatch* [6,11]. In this project, we employ a range of sensors to measure the dynamic response of a large Dutch highway bridge to varying traffic and weather conditions. When viewing this data (see Fig. 1a), one can easily distinguish various *transient events* in the signal that occur on different time scales. Most notable

P. Flach et al. (Eds.): ECML PKDD 2012, Part II, LNCS 7524, pp. 371–386, 2012.



(a) (b)

**Fig. 1.** (a) One day of strain measurements from a large highway bridge in the Netherlands. The multiple external factors affecting the bridge are visible at different time scales. (b) A detail of plot (a) showing one of the peaks caused by passing vehicles.

are the gradual change in strain over the course of the day (as a function of the outside temperature, which influences stiffness parameters of the concrete), a prolonged increase in strain caused by rush hour traffic congestion, and individual bumps in the signal due to cars and trucks traveling over the bridge. In order to understand the various changes in the sensor signal, one would benefit substantially from separating out the events at various scales. The main goal of the work described here is to do just that: we consider the temporal data as a series of superimposed effects at different time scales, establish at which scales events most often occur, and from this we extract the underlying signal components.

In this work, we approach the scale selection problem from a Minimum Description Length (MDL) perspective (see Section 3). The motivation for this is that we need a framework in which we can deal with a wide variety of representations for scale components. The MDL framework was shown to be sufficiently general to provide this flexibility by Hu et al. [3] for the problem of choosing the best model for a given signal. Our main assumption here is that separating the original signal into components at different time scales will simplify the shape of the individual components, making it easier to model them separately. Our results show that, indeed, these multiple models outperform (in terms of MDL score) a single model derived from the original signal. While introducing multiple models incurs the penalty of having to describe these multiple models, there are much fewer ‘exceptions’ to be described compared to the single model, yielding a lower overall description length. For instance, in the sensor data of Fig. 1a, cars are often passing in one direction while there is rush hour congestion in the opposite direction. Using multiple models, this is modeled accurately, while a single model will easily ignore these events.

The analysis of time scales in time series data is often approached from a *scale-space* perspective, which involves convolution of the original signal with Gaussian kernels of increasing size [12] to remove information at smaller scales. By subtracting carefully selected components of the scale-space, we can effectively cut up the scale space into $k$ ranges. In other words, signal processing offers methods for producing a large collection of derived features, and the challenge we face in this paper is how to select a subset of $k$ features, such that the original signal is decomposed into a set of meaningful components at different scales.

Our approach applies the MDL philosophy to various aspects of modeling: choosing the appropriate scales at which to model the components, determining the optimal number of components (while avoiding overfitting on overly specific details of the data), and deciding which class of models to apply to each individual component. For this last decision, we propose two classes of models representing the components respectively on the basis of a discretization and a segmentation scheme. For this last scheme, we allow three levels of complexity to approximate the segments: piecewise constant approximations, piecewise linear approximations, as well as quadratic ones. These options result in different trade-offs between model cost and accuracy, depending on the type of signal we are dealing with.

A useful side product of our approach is that it identifies a concise representation of the original signal. This representation is useful in itself: queries run on the decomposed signal may be answered more quickly than when run on the original data. Furthermore, the parameters of the encoding may indicate useful properties of the data as well.

The paper is organized as follows. Section 2 reviews the signal processing concepts used in this work and introduces the concept of scale-space decomposition. Section 3 shows how we encode the signal decompositions and use MDL to select the best subset of scales. Section 4 presents an empirical evaluation of our method on both real-world and artificial data. Section 5 links our method to related work. Finally, Section 6 states our main conclusions and ideas for future work.

## 2 Preliminaries

In this section we introduce the notation and the basic definitions used throughout the paper. In particular, we review the concept of the scale-space image of a signal and we show how to exploit it to define a set of candidate scale-space decompositions. We deal with finite sequences of numerical measurements (samples), collected by observing some property of a system with a sensor, and represented in the form of time series as defined below.

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

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

We also assume that all the considered time series have no missing values and that their sampling rate is constant.

### 2.1 The Scale-Space Image

The *scale-space image* [12] is a scale parametrization technique for one-dimensional signals[2] based on the operation of convolution.

[1] 32-bit floating point values in our experiments.

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

**Definition 2.** *Given a signal* $\boldsymbol{x}$ *of length* $n$ *and a response function (kernel)* $\boldsymbol{h}$ *of length* $m$*, the result of the* ***convolution*** $\boldsymbol{x} * \boldsymbol{h}$ *is the signal* $\boldsymbol{y}$ *of length* $n$*, defined as:*

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

In this paper, $\mathbf{h}$ is a Gaussian kernel with mean $\mu = 0$, standard deviation $\sigma$, area under the curve equal to 1, discretized into $m$ values.[3] Also, since $\mathbf{x}$ is finite, $\mathbf{x}[t-j]$ may be undefined. To account for these boundary effects, $\mathbf{x}$ is padded with $m/2$ zeros before and after its defined range. A complete overview on how to compute the Gaussian convolutions for discrete signals can be found in [7].

The convolution acts as a *smoothing filter* which smooths each value $\mathbf{x}[t]$ based on its surrounding values. The amount of removed detail is directly proportional to the standard deviation $\sigma$ (and thus $m$), 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 $\mathbf{x}$.

Given a signal $\mathbf{x}$, the family of $\sigma$-smoothed signals $\Phi_{\mathbf{x}}$ over scale parameter $\sigma$ is 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 $\sigma$, and $\Phi_{\mathbf{x}}(0) = \mathbf{x}$.

The signals in $\Phi_{\mathbf{x}}$ define a surface in the time-scale plane $(t, \sigma)$ known in the literature as the *scale-space image* [7,12]. This visualization gives a complete description of the scale properties of a signal in terms of Gaussian smoothing. Moreover, it has other properties useful for segmentation, as we will see later in the paper.

For practical purposes, the scale-space image is quantized across the scale dimension by computing the convolutions only for a finite number of scale parameters. More formally, for a given signal $\mathbf{x}$, we fix a set of scale parameters

$$S = \{2^i \mid 0 \leq i \leq \sigma_{max} \ \wedge i \in \mathbb{N}\}$$

and we compute $\Phi_{\mathbf{x}}(\sigma)$ only for $\sigma \in S$ where $\sigma_{max}$ is such that $\Phi_{\mathbf{x}}(\sigma)$ is approximately equal to the mean signal of $\mathbf{x}$.

As an example, Figure 2 shows the scale-space image of an artificially generated signal. The topmost plot represents the original signal, constructed by three components at different temporal scales: a slowly changing and slightly curved baseline, medium term events (bumps) and short term events (peaks). It is easy to visually verify that, by increasing the scale parameter, a larger amount of detail is removed. In particular, the peaks are smoothed out at scales greater than $\sigma = 2^4$, and the bumps are smoothed out at scales greater than $\sigma = 2^8$, after which only the baseline remains.

In the next section, we show how to manipulate the scale-space image to filter out the effects of transient events in a specific range of scales. This will lead to the definition of a signal decomposition scheme.

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

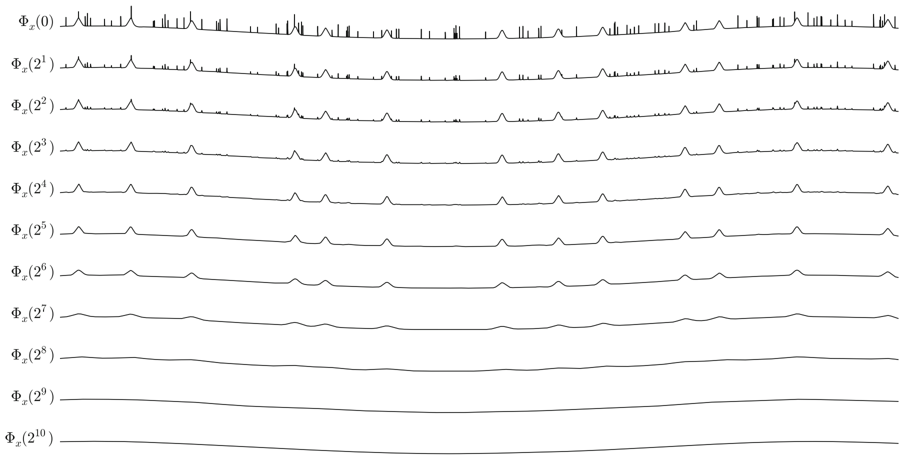


**Fig. 2.** Scale-space image of an artificially generated signal totalling 259200 points

### 2.2 Scale-Space Decomposition

Along the scale dimension of the scale-space image, short-time transient events in the signal will be smoothed away sooner than longer ones. In other words, we can associate to each event a maximum scale $\sigma_{cut}$ such that, for $\sigma > \sigma_{cut}$, the transient event is no longer present in $\Phi_{\mathbf{x}}(\sigma_{cut})$. This fact leads to the following two observations:

- Given a signal scale-space image $\Phi_{\mathbf{x}}$, the signal $\Phi_{\mathbf{x}}(\sigma)$ is only affected by the transient events at scales greater than $\sigma$. This is conceptually equivalent to a *low-pass filter* in signal processing.
- Given a signal scale-space image $\Phi_{\mathbf{x}}$ and two scales $\sigma_1 < \sigma_2$, the signal $\Phi_{\mathbf{x}}(\sigma_1) - \Phi_{\mathbf{x}}(\sigma_2)$ is mostly affected by those transient events present in the range of scales $(\sigma_1, \sigma_2)$. This is similar to a *band-pass filter* in signal processing.

As an example, reconsider the signal $\mathbf{x}$ and its scale-space image $\Phi_{\mathbf{x}}$ of Figure 2. Figure 3 shows (from top to bottom):

- the signal $\Phi_{\mathbf{x}}(0) - \Phi_{\mathbf{x}}(2^4)$, which is the result of a high-pass filtering; this feature represents the short-term events (peaks),
- the signal $\Phi_{\mathbf{x}}(2^4) - \Phi_{\mathbf{x}}(2^{10})$, which is the result of a band-pass filtering; this feature represents the medium-term events (bumps),
- the signal $\Phi_{\mathbf{x}}(2^{10})$, which is the result of a low-pass filtering; this feature represents the long-term trend.

Generalizing the example in Figure 3, we can define a decomposition scheme of a signal $\mathbf{x}$ by considering adjacent ranges of scales of the signal scale-space image. We formalize this idea below.

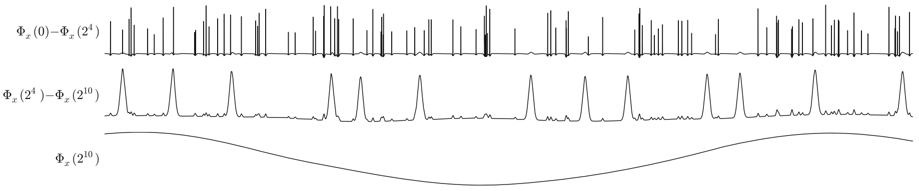


**Fig. 3.** Examples of signal decomposition obtained from the scale-space image in Figure 2

**Definition 3.** *Given a signal $\boldsymbol{x}$ and a set of $k-1$ scale parameters $C = \{\sigma_1, \ldots, \sigma_{k-1}\}$ (called the cut-points set) such that $\sigma_1 < \ldots < \sigma_{k-1}$, the* ***scale decomposition*** *of $\boldsymbol{x}$ is given by the set of component signals $D_{\boldsymbol{x}}(C) = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$, defined as follows:*

$$\boldsymbol{x}_i = \begin{cases} \Phi_{\boldsymbol{x}}(0) - \Phi_{\boldsymbol{x}}(\sigma_1) & \textit{if } i = 1 \\ \Phi_{\boldsymbol{x}}(\sigma_{i-1}) - \Phi_{\boldsymbol{x}}(\sigma_i) & \textit{if } 1 < i < k \\ \Phi_{\boldsymbol{x}}(\sigma_{k-1}) & \textit{if } i = k \end{cases}$$

Note that for $k$ components we require $k-1$ cut-points. This decomposition has several elegant properties:

- $\mathbf{x}_k$ can be seen as the baseline of the signal, as obtained by a low-pass filter;
- $\mathbf{x}_i$ for $1 \leq i < k$ are signals as obtained by a band-pass filter, and can be used to identify transient events;
- $\sum_{i=1}^{k} \mathbf{x}_i = \mathbf{x}$, i.e., the original signal can be recovered from the decomposition.

## 3 MDL Scale Decomposition Selection

Given an input signal $\mathbf{x}$, the main computational challenge we face is twofold:

- find a good subset of cut-points $C$ such that the resulting $k$ components of the decomposition $D_{\mathbf{x}}(C)$ optimally capture the effect of transient events at different scales,
- select a representation for each component, according to its inherent complexity.

As stated before, the rationale behind the scale decomposition is that it is easier to model the effect of a single class of transient events at a given scale than to model the superimposition of many, interacting transient events at multiple scales. We thus need to trade off the added complexity of having to represent multiple components for the complexity of the representations themselves. In this paper, we propose to use the Minimum Description Length (MDL) principle to approach this problem.

The Minimum Description Length [2] is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data. In our context, the two-part MDL principle states that the best model $M$ to describe the signal $\mathbf{x}$ 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 model,
- $L(\mathbf{x} \mid M)$ is the length, in bits, of the description of the signal when encoded with the help of the model $M$.

The possible models depend on the scale decomposition $D_{\mathbf{x}}(C)$ considered[4] and on the representations used for its individual components. An ideal set of representations would adapt to the specific features of every single component, resulting in a concise summarization of the decomposition and, thus, of the signal. In order to apply the MDL principle, we need to define a model $M_{D_{\mathbf{x}}(C)}$ for a given scale decomposition $D_{\mathbf{x}}(C)$ and, consequently, how to compute both $L(M_{D_{\mathbf{x}}(C)})$ and $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$. The latter term is the length in bits of the information lost by the model, i.e., the residual signal $\mathbf{x} - M_{D_{\mathbf{x}}(C)}$.

As the MDL framework is only applicable to discrete data, we first clarify below how we discretize the input signal $\mathbf{x}$ and all the subsequent operations. Subsequently, we will introduce the proposed representation schemes for the components and define the bit complexity of the residual and the model selection procedure.

### 3.1 Time Series Values Discretization

In order to use the MDL principle we need to work with a quantized input signal and scale-space image. Because of this, we assume that the values $v$ of both the input signal $\mathbf{x}$ and $\Phi_{\mathbf{x}}(\sigma)$, for each considered $\sigma$, have been quantized to a finite number of symbols by employing the function defined below:

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

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 $l = 256$. A similar approach is described in [3]. All the subsequent operations, from the computations of the scale decompositions to the encoding of the components, are kept in this quantized space.

### 3.2 Component Representation Schemes

Within our general framework, many different approaches could be used for representing the components of a decomposition. In the next paragraphs we introduce two such methods.

[4] Including the decomposition formed by zero cut-points ($C = \emptyset$), i.e., the signal itself.

**Discretization-Based Representation.** In some components of our data transient events always occur with similar amplitudes, mixed with long stretches of baseline values (see Figure 3). Hence, a desirable encoding could be one that captures this repetitiveness in the data by giving short codes to long stretches of the baseline and the commonly occurring amplitudes. Unfortunately, our original discretization is too fine-grained to capture regular occurrences of similar amplitudes. As a first representation, we hence propose to also consider more coarse-grained discretizations of the original range of values. We do this by discretizing each value $v$ in a component to a value $\lfloor Q(v)/2^i \rfloor$, where several values for $i$ are considered for each component, typically $i \in \{2, 4, 6\}$. By doing so, similar values will be grouped together in the same bin. The resulting sequence of integers is compacted further by performing run-length encoding, resulting in a string of $(v, l)$ pairs, where $l$ represents the number of times value $v$ is repeated consecutively. This string is finally encoded using a Shannon-Fano or Huffman code (see Section 3.3).

As a simplified illustration of how the MDL principle helps here to identify components, consider data generated by the expression $(67)^n(01)^n$ ($4n$ integers from the range $\{0, \ldots, 2^3 - 1\}$), where we assume $n$ and the range are fixed. In this data, each symbol occurs with the same frequency; we can encode the time series hence with $-\log_2(1/4) \cdot 4 \cdot n = 8n$ bits for the data, plus $8 \log n$ bits for the dictionary of frequencies. Consider now the decomposition of the signal into two time series, $6^{2n}0^{2n}$ and $(01)^{2n}$. The first component, of which the run-length encoding is $(6, 2n)(0, 2n)$, can be encoded using only 2 bits for the time series (as there is only one possible run-length value, we use 0 bits to encode the run-lengths), $8 \log n$ bits for the dictionary of amplitudes, and $3 \log n$ bits to identify the length of the one run-length ($\log n$ bit for identifying the number of run-lengths, in this case one, $\log n$ to identify the one run-length present, and $\log n$ to identify its frequency, from which the encoding with 0 bits follows). The second component can be encoded using $4n$ bits for the time series, as well as $8 \log n$ bits for the dictionary. Assuming we also use 1 bit per component to identify the type of encoding used, this gives us an encoding in $4 + 19 \log n + 4n$ bits. Comparing this to $8n + 8 \log n$ bits, for $n \geq 11$ we will hence correctly identify the two components in this simplified data.

**Segmentation-Based Representation.** The main assumption on which we base this method is that a clear transient event can be accurately represented by a simple function, such as a polynomial of a bounded degree. Hence, if a signal contains a number of clear transient events, it should be possible to accurately represent this signal with a number of segments, each of which represented by a simple function.

Given a component $\mathbf{x}_i$ of length $n$, let

$$z(\mathbf{x}_i) = \{t_1, t_2, ..., t_m\}, \quad 1 < t_i \leq n$$

be a set of indexes of the segment boundaries.

Let $\texttt{fit}(\mathbf{x}_i[a : b], d_i)$ be the approximation of $\mathbf{x}_i[a : b]$ obtained by fitting a polynomial of degree $d_i$. Then, we represent each component $\mathbf{x}_i$ with the approximation $\hat{\mathbf{x}}_i$, such that:

$$\begin{aligned} \hat{\mathbf{x}}_i[0 : z_1] &= \texttt{fit}(\mathbf{x}_i[0 : z_1], d_i) \\ \hat{\mathbf{x}}_i[z_i : z_{i+1}] &= \texttt{fit}(\mathbf{x}_i[z_i : z_{i+1}], d_i),\ 1 \leq i < m \\ \hat{\mathbf{x}}_i[z_m : n] &= \texttt{fit}(\mathbf{x}_i[z_m : n], d_i) \end{aligned}$$

Note that approximation $\hat{\mathbf{x}}_i$ is quantized again by reapplying the function $Q$ to each of its values.

For a given $k$-components scale decomposition $D_{\mathbf{x}}(C)$ and a fixed polynomial degree for each of its components, we calculate the complexity in bits of the model $M_{D_{\mathbf{x}}(C)}$, based on this representation scheme, as follows. Each approximated component $\hat{\mathbf{x}}_i$ consists of $|z(\mathbf{x}_i)| + 1$ segments. For each segment, we need to represent its length and the $d_i + 1$ coefficients of the fitted polynomial. The length $ls_i$ of the longest segment in $\hat{\mathbf{x}}_i$ is given by

$$ls_i = \max(z_1 \cup \{z_{i+1} - z_i \mid 0 < i \leq m\})$$

We therefore use $\log_2(ls_i)$ bits to represent the segment lengths, while for the coefficients of the polynomials we employ floating point numbers of fixed[5] bit complexity $c$. The MDL model cost is thus defined as:

$$L(M_{D_{\mathbf{x}}(C)}) = \sum_{i=1}^{k} (|z(\mathbf{x}_i)| + 1)\,(\lceil \log_2(ls_i) \rceil + c\,(d_i + 1))$$

So far we assumed to have a set of boundaries $z(\mathbf{x}_i)$, but we did not specify how to compute them. A desirable property for our segmentation would be that a segmentation at a coarser scale does not contain more segments than a segmentation at a finer scale.

The scale space theory assures that there are fewer zero-crossing of the derivatives of a signal at coarser scales [12]. In our segmentation we use the zero-crossings of the first and second derivatives.

More formally, we define the segmentation boundaries of a component $\mathbf{x}_i$ to be

$$z(\mathbf{x}_i) = \left\{ t \in \mathbb{R} \;\middle|\; \frac{d\mathbf{x}_i}{dt}(t) = 0 \right\} \bigcup \left\{ t \in \mathbb{R} \;\middle|\; \frac{d^2\mathbf{x}_i}{dt}(t) = 0 \right\}.$$

Figure 4b shows an example of segmentation obtained as above using fitted polynomials of degree 1.

However, many other segmentation algorithms are known in the literature [4,5] and all of them can be interchangeably employed in this context.

### 3.3 Residual Encoding

Given a model $M_{D_{\mathbf{x}}(C)}$, its residual $\mathbf{r} = \mathbf{x} - \sum_{i=1}^{k} \hat{\mathbf{x}}_i$, computed over the components approximations, represents the information of $\mathbf{x}$ not captured by the

[5] In our experiments $c = 32$.

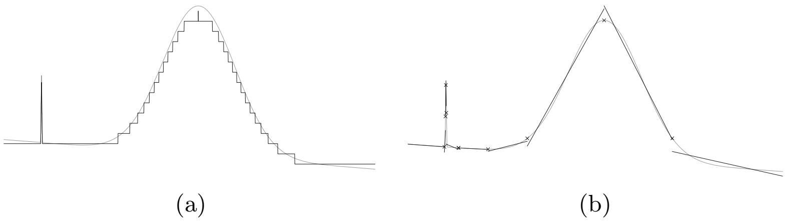


**Fig. 4.** Example of discretization-based encoding (a) and segmentation-based encoding with first degree polynomial approximations (the markers show the zero-crossings) (b)

model. Having already defined the model cost for the two proposed encoding schemes, we only still need to define $L(\mathbf{x} \mid M_{D_{\mathbf{x}}(C)})$, i.e., a bit complexity $L(\mathbf{r})$ for the residual $\mathbf{r}$.

Here, we exploit the fact that we operate in a quantized space; we encode each bin in the quantized space with a code that uses approximately $-\log(P(x))$ bits, where $P(x)$ is the frequency of the $x$th bin in our data. The main justification for this encoding is that we expect that the errors are normally distributed around 0. Hence, the bins in the discretization that reflect a low error will have the highest frequency of occurrences; we will give these the shortest codes. In practice, such codes can be obtained by means of Shannon-Fano coding or Huffman coding; as Hu et al. [3] we use Huffman coding in our experiments.

### 3.4 Model Selection

We can now define the MDL score that we are optimizing as follows:

**Definition 4.** *Given a model $M_{D_x(C)}$, its **MDL score** is defined as:*

$$L(M_{D_x(C)}) + L(\boldsymbol{r})$$

In the case of discretization-based encoding, the MDL score is affected by the cardinality used to encode each component. In the case of segmentation-based encoding the MDL score depends on the boundaries of the segments and the degrees of the polynomials in the representation. In both cases, also the cut-points of the considered decomposition affect the final score.

The simplest way to find the model that minimizes this score is to enumerate, encode and compute the MDL score for every possible scale-space decomposition and all possible encoding parameters. As we shall now show, this brute-force approach is practically feasible.

The number of possible scale decompositions depends on the total number of cut-points sets we can build from the computed scale parameters in $\Phi_{\mathbf{x}}$. We fix the maximum number of cut-points in a candidate set to some value $c_{max}$. This also means that we limit our search to those scale decompositions having $c_{max} + 1$ components or less. Moreover, given our wish to consider only simple approximations of the signals, we can also assume a reasonably low limit $d_{max}$

(in practice, $d_{max} = 2$) on the degree of the polynomials that approximate the segments of each given component.

Computing the MDL score for each encoded scale decomposition, obtained by ranging over all the possible configurations of cut-points $C_1, ..., C_{k-1}$, and all the possible configurations of polynomial degrees $d_1, ..., d_k$, hence requires calculating MDL scores for

$$\sum_{k=2}^{c_{max}+1} \binom{|\mathbf{S}|}{k-1} d_{max}^k$$

scale decompositions. This turns out to be a reasonable number in most practical cases we consider, and hence we use an exhaustive approach in our experiments.

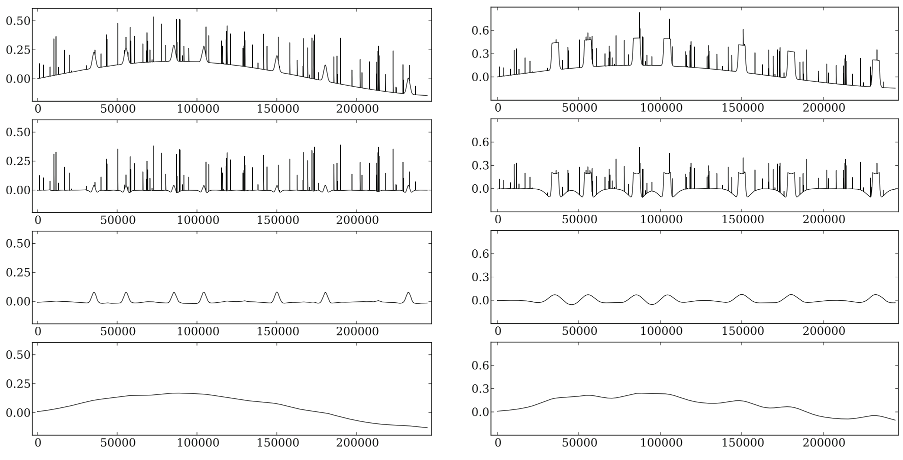
## 4 Experiments

In this section, we experimentally evaluate our method, both on artificial data and on actual sensor data from the highway bridge mentioned in the introduction. To evaluate the strengths and weaknesses of our method, we have tested it on a range of artificial datasets[6] that mimic some of the multi-scale phenomena present in the bridge data. Our constructed data deliberately varies from easy, with clearly separated scales, to challenging with a variety of event shapes and sizes. All artificial datasets represent sensor data measured at 1 Hz for a duration of three days (totaling 259,200 data points). The data was produced by combining three components at three distinct scales, resembling 1) individual events from vehicles, 2) traffic jams that last several tens of minutes, and 3) gradual change of the baseline, due to temperature changes of the bridge over the course of several days.

**Artificial Data.** We start by considering one particular dataset in detail (see Figure 5a). This dataset was constructed by using Gaussian shapes for both the small and medium-scale events, and a sine wave of period 2.25 days at the largest scale. Medium events have a constant height, whereas small-scale events have a random height. We limited the search space to decompositions having a maximum of 4 components (3 cut-points). As can be seen in Figure 5a, our method was able to identify the fact that this data contains three important scales. Furthermore, the method correctly identified the two necessary cut-points, such that the three original components were reconstructed. The selected cut-points[7] appear at scales $2^9 = 512$ and $2^{12} = 4096$. When considering the separated components in detail, some influence across the scale-boundaries is visible, for example where small effects of the 'traffic jams' appear among the small-scale

[6] The artificial datasets and the source code can be obtained by contacting the first author.

[7] Note that our method returns the boundaries between scales, rather than the actual scales of the original components.



(a) (b)

**Fig. 5.** Signals (top) and top-ranked decompositions for the two artificial datasets

events. These effects seem unavoidable, with the inherent limitations of the scale-space-based band-pass filtering and the discrete collection of scales we consider (powers of 2).

This optimal result has an MDL-score of 509,000 bits, being the sum of the model cost ($L(M) = 75,072$) and the error length ($L(D \mid M) = 433,928$). The second-ranked result on this data, with cut-points $C = \{2^{11}, 2^{13}\}$, shows a similar result, however with slightly more pronounced cross-boundary artifacts in the smallest scale, as is expected with a doubling of the lower cut-point. The MDL-score of this result is $64,896 + 450,487 = 515,383$. The $k = 1$ case, which corresponds to compression of the original signal without any decomposition, appears at rank three, with an MDL-score of $44,640 + 471,271 = 515,911$. This model obviously has a much lower model cost, due to having to represent only a single component, but this is compensated by the substantially higher error length, putting it below the scale-separated results. Ranks four and five represent two $k = 2$ results, where the former groups the small and medium scales together, and the latter the medium and large. All results in the top 10 relate to models that use polynomial representations ($d \leq 2$).

Not all artificial datasets considered produced perfect results. In Figure 5b, we show an example of a dataset that includes 'traffic jams' that resemble more closely some of the phenomena in the actual sensor data. In many cases, traffic jams appear fairly rapidly, and then show an increased load on the bridge over a prolonged period. This is modeled in the data by medium-scale events that start and stop fairly rapidly, and remain constant in the meantime. The best result found, with cut-points $C = \{2^{12}, 2^{13}\}$, is shown in Figure 5b. This demonstrates that the proposed method is not able to properly separate the medium and low-scale events. In fact, even though the medium component does identify the

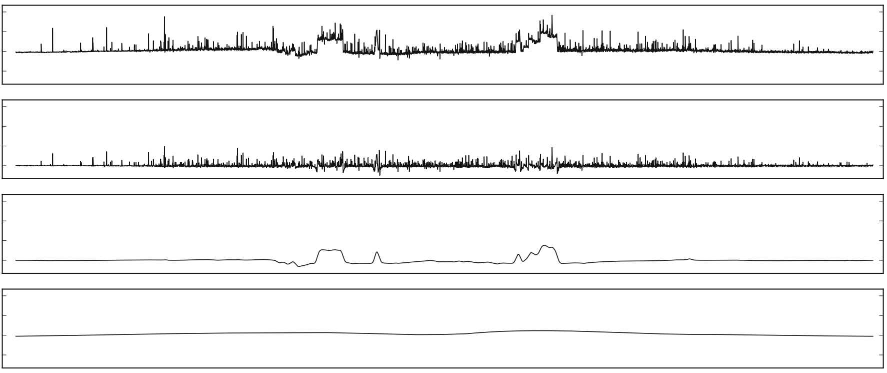


**Fig. 6.** Signal (top) and top-ranked scale decomposition for the InfraWatch data

location of the 'traffic jams', most of the rectangular nature is accounted for by the small scale. To some extent, this is understandable, as the start and end of the event could be considered high-frequency events with rapid changes in value. Therefore, parts of these events appear at a small scale, and the algorithm is mirroring this effect. In any case, the algorithm *is* able to identify the correct number of components, and is able to produce indications as to the location of the traffic jams. The top four results all show similar mixtures of scales, whereas the rank-five result groups the lowest two scales together. The $k = 1$ result appears at rank 14.

In order to better understand to what extent the proposed method is able to separate components at different scales, we carried out a more controlled experiment. We generated 11 different datasets constructed from 3 components. We fixed the scales of the short-term and long-term components respectively around $\sigma = 2^3$ and $\sigma = 2^{15}$, while the scale of the medium-term component varies from dataset to dataset in the range $(2^4, \ldots, 2^{14})$. The table below shows the number of components ($k$) of the top-ranked decomposition for the 11 datasets according to the scale parameter $\sigma$ of the medium-term component.

| $\sigma$ | $2^4$ | $2^5$ | $2^6$ | $2^7$ | $2^8$ | $2^9$ | $2^{10}$ | $2^{11}$ | $2^{12}$ | $2^{13}$ | $2^{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 1 | 1 | 1 |

As the table suggests, the proposed method fails to identify the right number of components when the scales are too close to each other. However, when the scales are separated sufficiently ($2^8 \leq \sigma \leq 2^{11}$), the right number of components is identified. Also in this case, all the top-ranked decompositions relate to models that use polynomial representations.

**InfraWatch Data.** As anticipated by the motivating example in the introduction, we consider the strain measurements produced by a sensors attached to a large highway bridge in the Netherlands. For this purpose, we consider a time

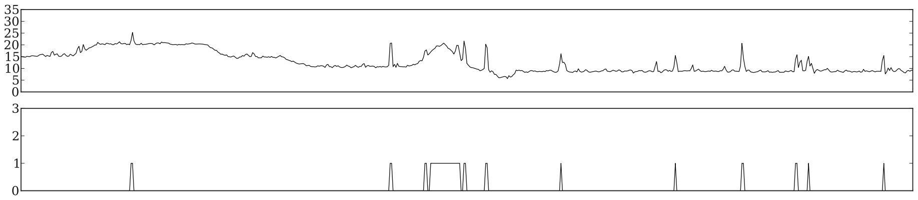


**Fig. 7.** A detail of the original strain signal (one hour) and the selected first component as represented with 4 symbols

series consisting of 24 hours of strain measurements sampled at 1 Hz (totaling $86,400$ data points). A plot of the data is shown in Figure 6 (topmost plot). We evaluated all the possible decompositions up to three components (two cut-points) allowing both the representation schemes we introduced. In the case of the discretization-based representations, we limit the possible cardinalities to 4, 16 and 64.

The top-ranked decomposition results in 3 components as shown in the last three plots in Figure 6. The selected cut-points appear at scales $2^6 = 64$ and $2^{11} = 2048$. All three components are represented with the discretization-based scheme, with a cardinality of respectively 4, 16, and 16 symbols. The decomposition has an MDL-score of $344,276$, where $L(M) = 19,457$ and $L(D \mid M) = 324,818$. The found components accurately correspond to physical events on the bridge. The first component, covering scales lower than $2^6$, reflects the short-term influence caused by passing vehicles and represented as peaks in the signal. Note that the cardinality selected for this component is the lowest admissible in our setting (4). This is reasonable considering that the relatively simple dynamic behavior occurring at these scales, mostly the presence or not of a peak over a flat baseline, can be cheaply described with 4 or fewer states without incurring a too large error. The middle component, covering scales between $2^6$ and $2^{11}$, reflects the medium-term effects caused by traffic jams. As in the artificial data, the first component is slightly influenced by the second one, especially at the start and ending points of a traffic jam. Finally, the third component captures all the scales greater than $2^{11}$, here representing the effect of temperature during a whole day. To sum up, the top-ranked decomposition successfully reflects the real physical phenomena affecting the data. The decompositions with rank 8 or less all present similar configurations of cut-points and cardinalities, resulting in comparable components where the conclusions above still hold. The first 2-component decomposition appears at rank 10 with the cut-point placed at scale $2^6$, which separates the short-term peaks from all the rest of the signal (traffic jams and baseline mixed together). These facts make the result pretty stable as most of the good decompositions are ranked first.

**An Application: Detecting Passing Vehicles.** The component selection and representation generated by the MDL procedure may be useful in itself for tasks such as classification. For example, consider the short-term component of the previous example, Figure 6 (second plot). It represents the traffic activity over the bridge and has been represented with a discretization-based scheme using 4 symbols. Figure 7 shows a detail (1 hour) of the discretized component (bottom) and the relative original signal (top). The first 2 symbols (0 and 1) respectively classify the absence or presence of a passing vehicle, while the other two, considerably less frequent, are outliers in the data. The represented component, as selected by MDL, can thus be used to monitor traffic activity over the bridge, a task that is considerably more challenging using the original signal, due to the variations introduced by temperature fluctuations and traffic jams.

## 5 Related Work

Papadimitriou et al. [9] propose a method to discover the key trends in a time series at multiple time scales (window lengths) by defining an incremental version of Singular Value Decomposition. In signal processing, Independent Component Analysis [1] aims at separating a set of signals from a set of mixed signals but, in its standard formulation, requires at least as many sensors as sources. Our method is able to operate on a single input sensor and a variable number of sources to be discovered. Megalooikonomou et al. [8] introduce a multi-scale vector quantized representation of time series which enables fast and robust retrieval. The considered scales are however predefined and our approach could be used as a preprocessing step to determine those to include in the dictionary. The Minimum Description Length principle has been applied to the problem of choosing the best representation for a given time series by Hu et al. [3]. The authors propose a method to choose the best representation (and its parameters) among APCA, PLA and DFT. While there are similarities with our method (we also use the MDL principle to select the best model parameters for a given component), the authors put the stress on discovering the intrinsic cardinality of the data, other than its constituent multi-scale components. MDL has also been adopted to detect changes in the distribution of a data stream by van Leeuwen et al. [10].

## 6 Conclusions and Future Work

We introduced a novel methodology to discover the fundamental scale components in a time series in an unsupervised manner. The methodology is based on building candidate scale decompositions, defined over the scale-space image [12] of the original time series, with an MDL-based selection procedure aimed at choosing the optimal one.

A useful side product of the presented technique, due to the adoption of MDL, is that each discovered component is represented independently according to its inherent complexity and often results in a cheaper model (in terms of MDL score) in relation to the original raw time series. These cheaper per-component representations may better serve tasks like classification, regression or association analysis for time series produced by inherently multi-scale physical and artificial systems.

We have shown that our approach successfully identifies the relevant scale components in both artificial and real-world time series, giving meaningful insights about the data in the latter case. Future work will experiment with diverse representation schemes and hybrid approaches (such as using combinations of segmentation, discretization and Fourier-based encodings). Moreover, another interesting research question is how to substitute the presently employed exhaustive search of the optimal decomposition with a computationally cheaper heuristic approach, which is necessary in the case of large time series data.

**Acknowledgments.** This work and the InfraWatch project are funded by the Dutch funding agency STW, under project number 10970. It was partially funded by a Postdoc grant from the Research Foundation-Flanders.

## References

1. Comon, P.: Independent component analysis, a new concept? Signal Processing 36(3), 287–314 (1994)
2. Grünwald, P.D.: The Minimum Description Length Principle. The MIT Press (2007)
3. Hu, B., Rakthanmanon, T., Hao, Y., Evans, S., Lonardi, S., Keogh, E.: Discovering the intrinsic cardinality and dimensionality of time series using mdl. In: Proceedings of ICDM 2011, pp. 1086–1091 (2011)
4. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. In: Data mining in Time Series Databases, pp. 1–22 (1993)
5. Keogh, E.J., Chu, S., Hart, D., Pazzani, M.J.: An online algorithm for segmenting time series. In: Proceedings of ICDM 2001, pp. 289–296 (2001)
6. Knobbe, A., Blockeel, H., Koopman, A., Calders, T., Obladen, B., Bosma, C., Galenkamp, H., Koenders, E., Kok, J.: InfraWatch: Data Management of Large Systems for Monitoring Infrastructural Performance. In: Cohen, P.R., Adams, N.M., Berthold, M.R. (eds.) IDA 2010. LNCS, vol. 6065, pp. 91–102. Springer, Heidelberg (2010)
7. Lindeberg, T.: Scale-space for discrete signals. IEEE Trans. Pattern Analysis & Machine Intelligence 12(3), 234–254 (1990)
8. Megalooikonomou, V., Wang, Q., Li, G., Faloutsos, C.: A multiresolution symbolic representation of time series. In: Proceedings of ICDE 2005, pp. 668–679 (2005)
9. Papadimitriou, S., Yu, P.: Optimal multi-scale patterns in time series streams. In: Proceedings SIGMOD 2006, pp. 647–658. ACM (2006)
10. van Leeuwen, M., Siebes, A.: StreamKrimp: Detecting Change in Data Streams. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 672–687. Springer, Heidelberg (2008)
11. Vespier, U., Knobbe, A., Vanschoren, J., Miao, S., Koopman, A., Obladen, B., Bosma, C.: Traffic Events Modeling for Structural Health Monitoring. In: Gama, J., Bradley, E., Hollmén, J. (eds.) IDA 2011. LNCS, vol. 7014, pp. 376–387. Springer, Heidelberg (2011)
12. Witkin, A.P.: Scale-space filtering. In: Proceedings IJCAI 1983, San Francisco, CA, USA, pp. 1019–1022 (1983)