Finding Intuitive Abstractions of High-Frequency Data

Silvia Miksch, Andreas Seyfang

Institute of Software Technology, University of Technology, Vienna,
{silvia,seyfang}@ifs.tuwien.ac.at

Abstract. In this paper we describe ways to transform a curve constituted by a series of data points delivered by monitoring devices into a series of bends and lines between them. The resulting qualitative representation of the curve can easily be utilized in a Knowledge Based System.
Since the representation obtained resembles the way humans describe curves, e.g. "this bend is not sharp enough to be normal", the transformation of the data described in this paper promises to facilitate knowledge acquisition in the field of interpreting high-frequency data in the medical domain.
Our analysis performs best on curves that are periodical but too irregular to be analyzed by Fast Fourier Transformation or other standard methods.

1 Introduction

In all fields of medicine one is confronted with rhythmical data. By rhythmical we mean data which shows repeated patterns which slightly vary from instance to instance but still have enough in common to make it interesting to compare them, like ECG.

If such patterns are strongly regular, they can easily be analyzed by Fast Fourier Transformations (FFT) [2], which are a widely used and fairly exploited method. The result of such a transformation is a spectrum of frequencies, from which each oscillation is composed. While meaningful in some fields of applications, like music or signal processing, this type of information by itself is not meaningful for medical experts in most cases. Furthermore, the data available in medical high-frequency domains, like in artificial ventilation of neonates, is rarely regular enough to yield useful results when analyzed by FFT.

In many domains, like ECG, there is a long tradition in analyzing graphs and thus a lot of – in part informal – knowledge about the appearance of a graph and the health status of the corresponding patient. The way a graph appears to a human depends on the fashion of the bends it makes (sharp or smooth), the direction and straightness of the lines between them (steep, flat, up, down), and the relative position of characteristic points in the graph within one oscillation.

These types of characteristic features are far away from conventional tools for the analysis of oscillating data, since they focus only on the mathematical aspects of the data like frequencies or other highly abstract parameters. It is nearly impossible to transform the experiences of human experts in analyzing a graph in their mind and the way they formulate their constraints into such a mathematical set of parameters.

To bridge this gap, we developed a method to abstract characteristics similar to those used by human experts from a graph. In particular, we decompose the graph into a sequel of repeated patterns. Each is described by a set of bends and lines in between. A bend is placed in the graph where it changes its direction. It has a "sharpness" defining how rapid the change takes place and a position. A line is place between each pair of bends to represent the points in between. Its important feature is the inclination it shows.

Of course both the amount of sharpness of a bend necessary to consider it significant and the minimum distance of neighboring bends required to distinguish them from noise will vary from person to person and from case to case. Still there is consensus on most of the important cases.

These abstracted characteristics can be visualized as bar charts or graphs. They can also be used to match the graphs with the conditions of rules in a knowledge base like "If the ascent of the first line exceeds that of the third then ..." or "If the distance of the 2^{nd} and 3^{rd} corner decrease by more than 50 % during the first minute of measurement, then ...".

Thus, existing knowledge about the interpretation of graphs can be utilized with significant less effort on information transformation compared to the use of conventional tools which require highly abstract input.

Of course, such abstractions can only be done retrospectively, or at least with a significant delay when done on-line. Lacking data from ICUs at the moment (for technical reasons), we demonstrate our algorithms on data from ergonomic studies in rowing which show the same characteristics as the clinical data for which the algorithms were intended.

In section 2 we show how our approach differs from other work. In section 3 we explain our approach in depth. In section 4 we describe the obtained data and how to use this data to bridge the gap towards knowledge-based systems. In section 5 we list the benefits of our approach.

2 Related Work

On-line monitoring at Intensive Care Units (ICUs) produces a high volume of high-frequency data generated by monitoring devices. These data need to be analyzed and interpreted to reduce the information overload of the medical staff and to guarantee quality of patient care. The interpretation of time-series is a very challenging task. The temporal properties are very important aspects in the medical domain, particularly when dealing with the interpretation of continuously assessed data. The most common methods are time-series analysis techniques [1], control theory, probabilistic or fuzzy classifiers. However, these approaches have a lot of shortcomings, which lead to apply knowledge-based techniques to derive qualitative values or patterns of the current and the past situations of a patient, called *temporal data abstraction*. Several significant and encouraging approaches have been developed in the past years (e.g., $TrenD_x$ [3], RÉSUMÉ [9, 10], VIE-VENT [7], Larizza et al. [6], Keravnou [5]).

These approaches rely on predefined qualitative descriptions or categories of temporal data abstractions. For example, the RÉSUMÉ project [9, 10] recommends to apply state, gradient, rate, and simple pattern abstractions, Larizza et al. [6] are using basic and complex abstraction, and the temporal data abstraction module in the VIE-VENT system [7] tries to arrive at unified, context-sensitive qualitative descriptions applying smoothing techniques of data oscillating and expectation-guided schemata for trend-curve fitting.

However, we are going one step back and want to explore, which kinds of temporal data abstractions are need for rhythmical data. Therefore, we are not presenting various temporal data abstraction methods for longitudinal data. We are demonstrating a way to acquire complex data abstraction methods to arrive at qualitative descriptions, like in an ECG trace "the peak of the P-wave is increasing significantly between 400 and 700 msec". A similar technique is the "Time Series Workbench" [4], which approximates data curves with a series of line-segments. However, we are going beyond the approximation by line-segments and take the particular characteristics of a graph into account, like the "sharpness" of a curve.

3 The Algorithm

While mathematicians might be horrified by the notion of a graph being a series of bends connected by rather straight lines this resembles the cognitive model most non-mathematicians use when looking at a graph. But how can we find a formal definition of such an informal entity as a bend?

There are two indications for bends in a curve: First, the second-order derivative of the curve shows an minimum in places where the original curve does a "bend to the right", i.e. changes from increase to decrease, and a maximum, where the original curve does a "bend to the left", i.e. changes from decrease to increase.

Second, we calculate linear regressions for a time window sliding over the curve as described in [8]. In places where the curve shows a bend, reducing the length of the interval will lead to a decrease in the standard error of the resulting linear regression. In places where there is no significant bend, shortening the time window will not decrease the standard error.

We will first explain both approaches in detail and then discuss which of them is more suitable for which type of data.

3.1 Using the Changes of the Derivative

Figure 1 shows an abstract example. A bend in the curve is a synonym to a change in its derivative. The bigger the change in the derivative, the sharper the bend – and the bigger the absolute value of the second-order derivative.

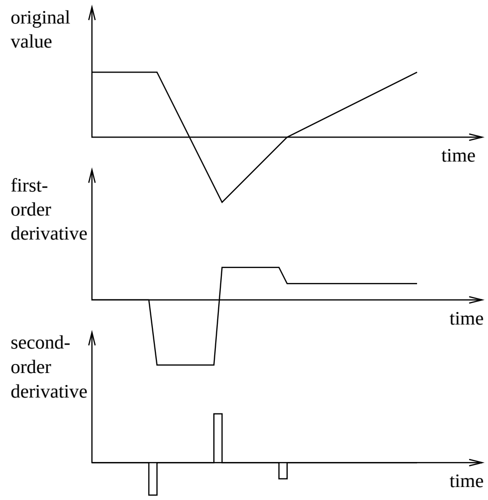

Fig. 1. Abstract demonstration of bends in the original graph and its second-order derivative: In places where the original graph shows a bend, the first-order derivative's changes, which causes a peak in the second-order derivative.

While this notion is perfectly true for small derivatives, looking at changes in the absolute value of the derivative will overemphasize relatively small changes in places of high derivative. If e.g. a derivative of 10 changes by 2, this might not seem too significant to a human spectator while a change from 0 to 2 certainly will. The second-order derivative is 2 in both cases. So its value will not reflect the users estimations. Figure 2 shows an example of a peak in the second-order derivative where a human would not see a significant change.

Turning to relative changes in the derivative only works for steep slopes and will overemphasize changes in flat regions of the curve. Instead, we are using the angle of the derivative. So instead of the derivative itself we calculate the angle α as $\tan \alpha = \frac{\Delta y}{\Delta x}$ and use the derivative of this function as indicator for significant changes in the curve.

Figure 3 shows an example, where this function nicely reflects human perspective. The curve slightly but constantly turns up. So it is difficult to say, where a single corner should be. The derivative of the derivative's angle (i.e. the angle of inclination of the original curve) is constantly but slightly increasing at that part of the curve, resembling the humans indecision.

In practical applications, calculating the derivative as the difference in the y-coordinate of two neighboring data points (divided by the difference in their x-coordinate) does not work on noisy input data, because the small erroneous oscillations of the curve might result in the derivative oscillating enough to hide the overall tendency of the curve. Comparing each point with the point following n points later instead of the ultimate neighbor (and placing the result in the middle between the two points) often yields sufficiently smooth graph for the

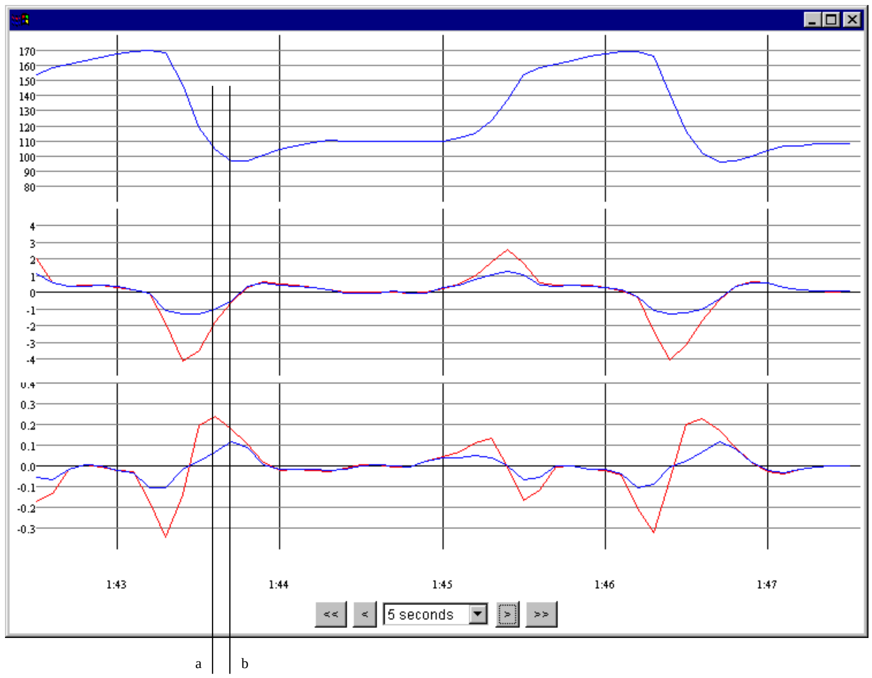

Fig. 2. The topmost graph shows the original data. In the middle the gray graph (that with the bigger extrema) shows the absolute value of the first derivative and the black graph shows the angle of inclination. At the bottom, the derivatives of both graphs in the middle are shown. (The more moderate, black one is the derivative of the moderate one in the middle). While the change in absolute value of the first-order derivative is biggest in (a), the change in the angle associated with the derivative is biggest in (b). Human spectators seem to prefer (b) over (a) if asked to define the significant corner at this portion of the graph.

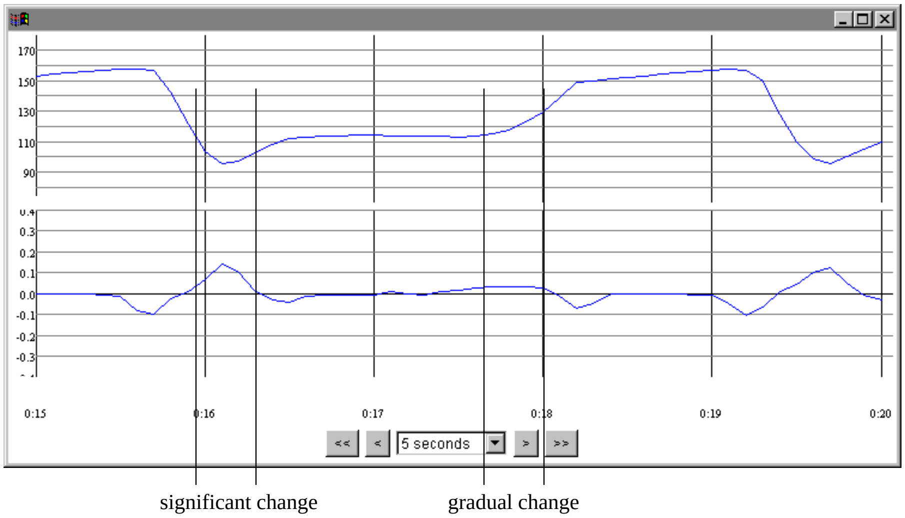

Fig. 3. At gradual turns of the original curve (at the top), were a human spectator is pressed hard to point at any exact position of a single corner, the indication function (below) is trapezoidal resembling the humans indecision.

derivative without the need to smoothen the original curve. The number of intermediate points n should be bigger then the typical wave length of erroneous oscillations or – for nondeterministic noise – simply big enough to suppress the portion of the noise in the result of

$$calculated\,derivative = \frac{n * real\ derivative + noise}{n} = real\ derivative + \frac{noise}{n}$$

where *noise* is the average distant between a measured value and the real value, *real derivative* is the derivative of the ideal graph drawn from the real values (which is not known, of course) and *calculated derivative* is the value resulting from this calculation.

3.2 Using the Length of the Regression Line

The algorithm presented in the following seeks to detect bends in the graph by first calculating a linear regression for short sections of the graph and then checking whether reducing the size of the section reduces the standard error of the regression line.

The reason for applying linear regression lies in its ability to give an abstract representation of a set of data points and at the same time an estimate, how will this abstraction represents the actual data (by the standard error). If the regression line does not fit to the curve because it make a bend, then cutting the ends of the line results in a significantly reduced standard error. If the regression line does not fit the curve because the curve is noisy, a shorter regression line will have an equally high standard error as the original (full length) one. This distinction can be exploited to detect bends in a graph.

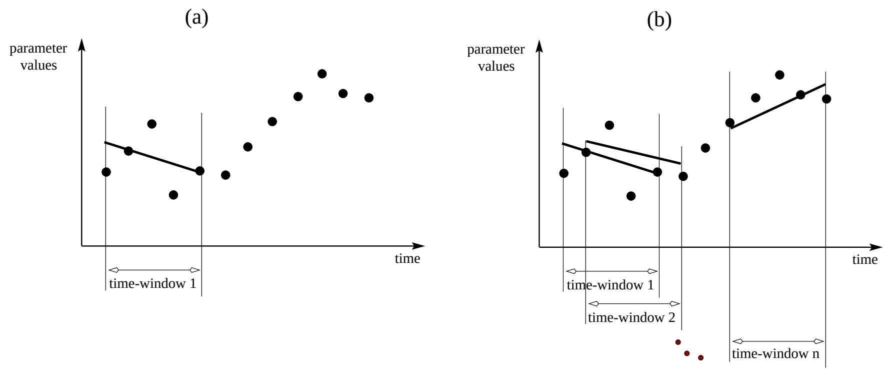

Fig. 4. The calculation of the linear regression is done for a time window of fixed size sliding over the entire curve in small steps. (a) shows a single time window and the line calculated from the data points within it. (b) shows a sequence of overlapping time windows and the resulting lines.

As illustrated by figure 4, we slide a window of consideration (*time window*), which is of fixed size over the time axis of the curve in small steps. For each instance of the time window, we calculate a linear regression for the data points contained in it. As opposed to [8], for this application the step width should equal the rate of data points (if there is one measurement per second, step width should be one second) and the length of the time window should be short enough to follow significant bends but much longer than erroneous oscillations.

So, for example, if the sampling rate is 1 measurement per second and the oscillations caused by noise show a wave length of up to 5 seconds, the step width will be one and the size of the time window will be between about 7 and 10 seconds. We will thus receive one regression line per data point, calculated from its 7 to 10 neighbors.

The standard error of a linear regression line shows, how well the points, which are represented by that line fit together respectively to that line. The bigger the average distance, the bigger the standard error.

For each regression line, we take a look at its ends (see figure 5). On each end, there might be some neighboring points on the same side of the line. If a smooth curve takes a bend they will be numerous, if the graph is rather straight, but oscillating around the line, there will be very few points at the same side of the line.

Next we shrink the time window to skip those groups of points on both ends which altogether lie on the same side of the curve and recalculate the linear regression for this second, smaller time window. If the distance of the skipped points exceeds the average distance of all points in the (first) time window to

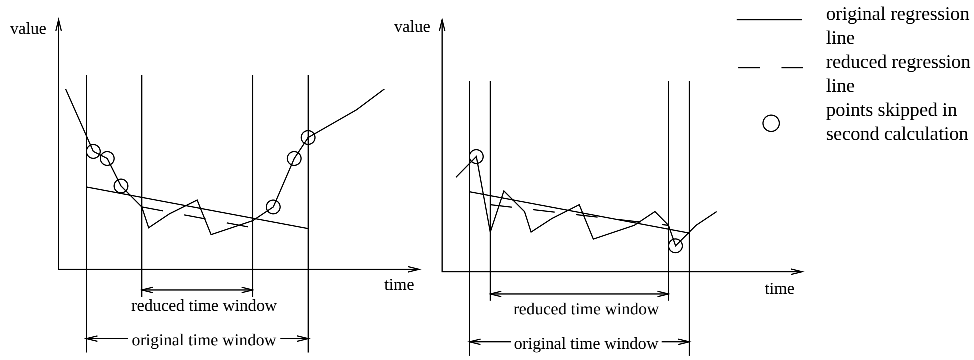

Fig. 5. If the graph shows a bend in the interval under consideration (example on the left-hand side), there is a considerable number of data points on each end of the regression line which lie on the same side. Skipping them in the recalculation of the regression reduces the standard error to which the skipped points contributed significantly. If there is no bend (example on the right-hand side), skipping the few points on the ends does not reduce the standard error.

the (first) regression line, the standard error of the second regression line will be smaller than that of the first one. In this case we can assume that the deviation of the points on the ends of the line are not just an incident, but caused by a bend in the curve.

The difference in length between the first and the second time window as well as the decrease of the standard error are measures for the "sharpness" of the curve. Thus both of them can be used as indication function. Both only give positive values. The direction of the curve can be derived from the side of the regression line, on which the skipped data points lie. So we assign minus to bends to the right and plus the bends to the left and supply the absolute value of the indication function with this sign to produce an indication function compatible with the one described in section 3.1.

3.3 Common Issues of Both Approaches

In both cases (second-order derivative and length of the regression line) a bend in the curve will not yield only one high value at a single position on the time axis, but a more or less significant peak. Especially, bends with bigger radius result in a series of peaks or a long flat "hill" in the second-order derivative respectively a "valley" in the curve showing the length of the regression line.

To suppress such concurring peaks one can simply define a minimum distance (along the time axis) and only chose the highest peak out of several of them if their distance is below this threshold.

A better way is to consider both of two neighboring peaks only if they are separated by a local minimum of a certain depth. To see the difference to the above strategy consider the following cases: First, two sharp bends close to each other and second, a long slight bend.

Two sharp bends produce two high peaks with a clearly distinguishable minimum (of the absolute value) in between. If you only consider the distance on the time axis of the two peaks, you will have to ignore one of them, if you consider the minimum between them, you will accept both peaks to be significant.

A long slight bend results in a series of peaks with nearly no minimum between them. If you consider the distance along the time axis, the first and the last minor peaks might be far enough from each other to let both of them seem justified. If you look at the minima between them, you will ignore all but one of them.

Many curves of real data show small opposite bends which should be considered as a single straight line. A small threshold for the absolute value of the indication function does this job.

3.4 Comparing the Two Approaches

The first approach – the change in the angle of inclination – is very intuitive if applied on smooth graphs. Applied on noisy input data, the graph of its indication function can get too distorted to be usable.

The second approach – the length of the regression line – is harder to compute than the first one. The outstanding advantage of linear regression is that it minimizes the influence of noise on the result. If the original graph shows numerous random peaks, they can fool the second algorithm because they might inhibit proper reduction of the regression line.

In such cases, a combination of both approaches performs best: The indication function is the change in the ascent of the regression lines.

1. The regression lines are calculated as described in section 3.2.
2. For each of them, the angle of inclination is calculated.
3. Then the resulting values are merge to a new function (replacing the first derivative in the first approach)
4. The derivative of this function is calculated as the indication function for detecting bends.

To summarize, given smooth input data, the first approach performs better. The more smoothing is necessary before or while calculating the derivative, the small this gain becomes.

4 The Output

4.1 The Extracted Characteristics of a Graph

As results of the transformation of the discrete data points into bends and lines we receive three sequels of different types of data: The bends, the corners of the original curve at those places where bends were detected, and the lines between the bends.

Bends Each bend is described by the position of its middle along the time axis, the height of the corresponding peak in the indication function (second-order derivative or length of regression line) and the area of the peak measured from one intersection with the zero-line to the next. Figure 6 illustrates this calculation for one bar, while figure 7 gives a practical example.

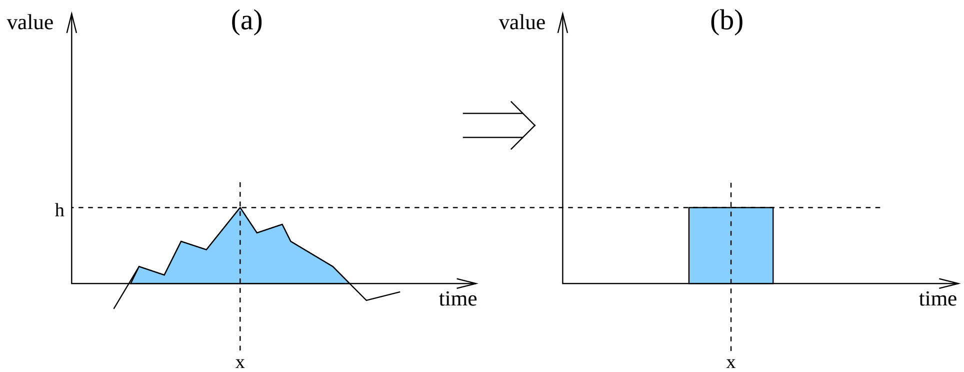

Fig. 6. The height and area of the bar (b) representing a bend equal the height and area (grayed) of the corresponding peak (a) in the indication function, the width of the bar is the quotient of area and height.

Corners The x-coordinate of the corner clearly equals the middle of the bend. The y-value can be the y-coordinate of the nearest point in the original curve. To reduce influence of noise, it is necessary to take the average of some of its neighbors into account too in most cases. Integrating too many of them in the calculation will distort the result towards the inner side of the bend.

The Line in Between The lines between the bends can either be drawn just as connections of the corners of the curve, or they are calculated as a linear regression of the points of the original curve between the bend.

4.2 Interfacing Knowledge-Based Systems

To bridge the gap currently observed between data analysis and knowledge based systems (KBS), we transform the output into clauses compatible with those use by a KBS. In a first step, we gather the absolute values for each characteristic. Then, we calculate the relative values (e.g. distance of corners). Finally, we transform these quantitative values into qualitative ones via tables containing the qualitative ranges for each parameter.

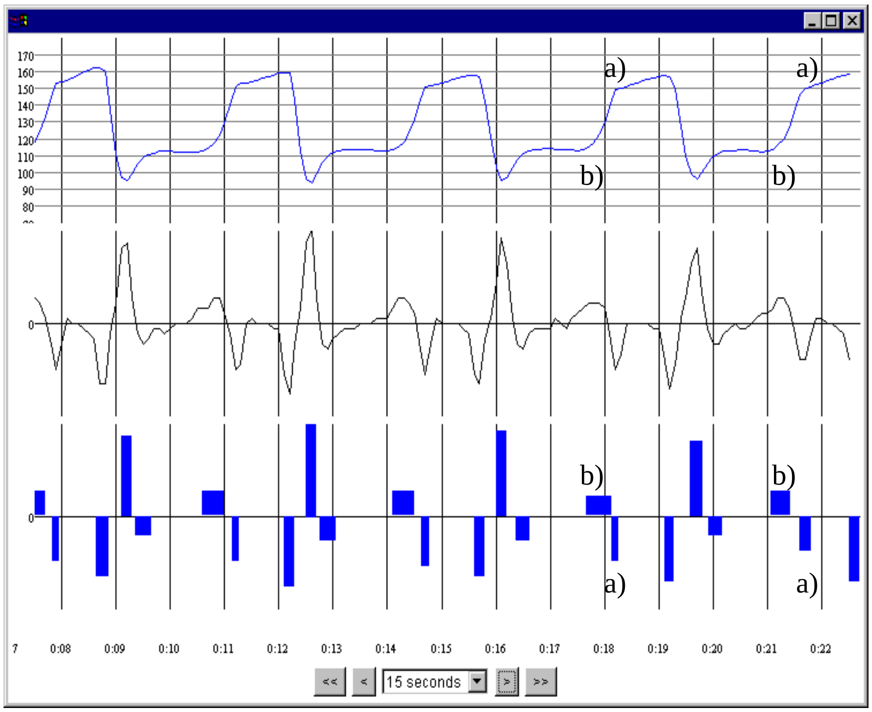

Fig. 7. Starting at the top, we show the original graph, the indication function and the bars representing the significant bends. a) shows an example of an irregularity in the curve which a human could also dedect if concentrating on every detail: the bend in the right-most oscillation is not as sharp as the corresponding one in the other oscillations. b) draws our attention to a feature not perceptible by looking at the raw data: the long-spread bow to the left of the second oscillation from the right is not as sharp as the others as indicated by inferior height of the bar. The corresponding part of the original curve does not seem different by itself. The significance of the features found in a) and b) depends on the domain knowledge about the data represented by the curve.

Absolute Values For each bend, the above abstraction yields the following data:

- The corresponding maximum in the indication function shows how sharp the bend is.
- The area of the corresponding peak shows how big (in degrees) the change of the original graph is.
- The x–coordinate shows the position of the corner on the time–axis.

For each line, the inclination of is computed.

Relative Values Each of the above values is measured against the average of previous instances in an interval of time defined by the user. The deviation is given both absolute and relative.

Qualitative Values The quantities computed in the two steps above are qualified using a set of tables. The tables are grouped in a default hierarchy so for each of it, it is possible to define special values, but similar tables need not be entered multiple times.

5 Benefits

The abstraction of characteristics from a stream of raw data points offers the following opportunities:

Compact visualization Displaying only the important features of a graph in an abstract form instead of the original graph allows for easy detection of trends and outliers which otherwise would be buried in the overwhelming impression of countless oscillations.

New Insights Plotting the representations of the features together with the conventional graph of the curve draws the interest of the user to irregularities not visible in the original graph without the need to be familiar with the new representation beforehand. Thus, new insights can be gained by comparing the conventional representation of the graph, the displayed features of some characteristics and the underlying case (background information).

Bridge to Knowledge Representation The abstracted characteristics can be matched against conditions from a rule base. So the curves can be tagged according to a set of classifications stored in a knowledge base. This aspect is crucial for the integration of high-frequency data and symbolic systems such as the sceletal planning system implemented in the Asgaard project.

6 Conclusion

We have presented several methods to capture complex rhythmical curves by transforming them into a sequel of bends, corners, and lines, based on the observation that a bend in the curve is synonym to a change in its inclination. These feature vectors resemble the human understanding of repeating patterns in an intuitive way.

Previous approaches in the field of temporal data abstraction focus on predefined qualitative descriptions or categories. We explore different ways to extract particular characteristics of a curve, which will lead to appropriate qualitative descriptions.

Our approach is applicable to data where Fast Fourier Transformation fails because the oscillations is not regular enough for such a strictly numerical algorithm. Furthermore, a frequency spectrum is a less intuitive representation of a curve than a set of corners and lines in many medical domains.

Acknowledgments

We thank Klaus Hammermüller for supplying the data from the ergonomic studies and Werner Horn, Christian Popow, Robert Kosara, and Georg Duftschmid for their useful comments. This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung - FWF" (Austrian Science Foundation), P12797-INF.

References

1. R.K. Avent and J.D. Charlton. A critical review of trend-detection methologies for biomedical monitoring systems. *Critical Reviews in Biomedical Engineering*, 17(6):621–659, 1990.
2. J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
3. I. J. Haimowitz and I. S. Kohane. Managing temporal worlds for medical tread diagnosis. *Artificial Intelligence in Medicine, Special Issue Temporal Reasoning in Medicine*, 8(3):299–321, 1996.
4. J. Hunter. Knowledge-based interpretation of time series data from the neonatal icu. presentation, 1998.
5. E. T. Keravnou. Temporal abstraction of medical data: Deriving periodicity. In N. Lavrac, E. Keravnou, and B. Zupan, editors, *Intelligent Data Analysis in Medicine and Pharmacology*, pages 61–79. Kluwer Academic Publisher, Boston, 1997.
6. C. Larizza, R. Bellazzi, and A. Riva. Temporal abstractions for diabetic patients management. In *Proceedings of the Artificial Intelligence in Medicine, 6th Conference on Artificial Intelligence in Medicine Europe (AIME-97)*, pages 319–30, Berlin, 1997. Springer.
7. S. Miksch, W. Horn, C. Popow, and F. Paky. Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants. *Artificial Intelligence in Medicine*, 8(6):543–576, 1996.

8. S. Miksch, A. Seyfang, W. Horn, and Popow C. Abstracting steady qualitative descriptions over time from noisy, high-frequency data. In W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, editors, *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AIMDM'99*, Berlin, 1999. Springer. in print.
9. Y. Shahar. A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90(1-2):267–98, 1997.
10. Y. Shahar and M. A. Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine, Special Issue Temporal Reasoning in Medicine*, 8(3):267–98, 1996.