Abstraction and Representation of Repeated Patterns in High-Frequency Data

Silvia Miksch[1], **Andreas Seyfang**[1] and **Christian Popow**[2]

Abstract. Monitoring devices in intensive care units deliver enormous streams of data in the form of snapshot measurements. In contrast, physicians use high-level abstractions in reasoning about the parameters observed.

Standard data abstraction algorithms can only handle data which are more regular than many variables observed in medicine. A typical example is the ECG in intensive care, where electric currents are measured at the skin surface and displayed amplified in order to detect problems in the conduction system of the muscular contraction pattern.

We developed an algorithm to transform a curve constituted by a series of data points into a set of bends and lines in between them. The resulting qualitative representation of the curve can be expressed as a list of objects each describing a bend. In this format, it can easily be utilized in a Knowledge-Based System (KBS).

In addition, in the case of rhythmical data, comparing selected bends in all cycles of the oscillation yields new information. This comparison can be done by plotting derived data as separate graph beside the original one or by encoding the knowledge behind the reasoning in rules in the KBS.

Our algorithm performs best on curves which are rhythmical but too irregular to be analyzed by Fast Fourier Transformation or other standard methods aiming at describing regular patterns.

We demonstrate our approach by displaying heart rate and Q-S-distance graphically aside of ECG-data (to detect impeded conduction) and by showing example code for rules detecting pathological deviations from the standard based on the qualitative representation of the curve.

1 Introduction

In all fields of medicine one is confronted with rhythmical[3] data. By rhythmical we mean data which show 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) [6], a widely used and fairly exploited method. The result of such a transformation is a spectrum of frequencies, describing the harmonics of an oscillation. While meaningful in some fields of applications, like music or signal processing, this type of information by itself may not be meaningful for medical experts because of the complexity of the results and the post hoc type of analysis and because the data available in medical domains, are 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 an expert depends on the kind and pattern of the bends it makes (sharp or smooth), the direction and straightness of the lines in between them (steep, flat, up, down), and the relative position of characteristic points in the graph within one oscillation cycle.

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 series of repeated patterns. Each pattern is described by a set of bends and lines in between. A bend is a (typically short) section of the graph where it changes its direction. It has a position and a "sharpness" defining how rapid the change takes place. A line is placed in between each pair of bends in order to represent the data points in between. Its most important feature is its inclination.

There are two characteristics of a bend, its sharpness – which is necessary to consider it significant – and the minimum distance of neighboring bends – which is required to distinguish them from noise. 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 decreases 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

[1] Institute of Software Technology, Vienna University of Technology Favoritenstraße 9-11/188, A-1040 Vienna, Austria, email: {silvia,seyfang}@ifs.tuwien.ac.at

[2] Department of Pediatrics, University of Vienna Währinger Gürtel 18-20, A-1090 Vienna, Austria, email: popow@akh-wien.ac.at

[3] In this paper, we denote data as rhythmical, if they exhibit repeated patterns, but is not regular enough to be considered periodical in the sense of signal processing.

transformation compared to the use of conventional tools which require highly abstract input.

Of course, such abstractions can be done retrospectively at best. If on-line, a significant delay between the time of measurement and the time of calculation since considering a sufficiently large time interval is indispensable for thorough analysis.

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 application of the generated data for both building a bridge between monitoring and knowledge-based systems on the one hand and to give the user compact information on the other hand.

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 [2], 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., $TrenD_x$ [9], RÉSUMÉ [18, 19], VIE-VENT [16], Larizza et al. [13], Keravnou [12], Belazzi [3]). A comprehensive selection of various approaches in intelligent data analysis in medicine can be found in [14].

These approaches rely on predefined qualitative descriptions or categories of temporal data abstractions. For example, the RÉSUMÉ project [18, 19] recommends to apply state, gradient, rate, and simple pattern abstractions, Larizza et al. [13] are using basic and complex abstraction, and the temporal data abstraction module in the VIE-VENT system [16] tries to arrive at unified, context-sensitive qualitative descriptions applying smoothing techniques of data oscillating and expectation-guided schemata for trend-curve fitting. In contrast, Calvelo et al. [4] seek to separate stable patients at an adult ICU from such in a critical situation by applying machine learning methods.

A comprehensive study about various approaches of intelligent data analysis for medical diagnosis using machine learning and temporal abstraction techniques can be found in [15]

However, we are going one step back and want to explore, which kinds of temporal data abstractions are needed for rhythmical data. We are demonstrating a way to acquire complex data abstraction methods to arrive at qualitative descriptions, like "the variability of the $\overline{PQ}$-distance increase significantly during the last 2 hours" which directly indicate medically relevant facts like – in this case – a problem in the excitation conduction from the atria to the ventricles.

A similar technique is the "Time Series Workbench" [11], 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 a 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 [17]. 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 characterized by 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.

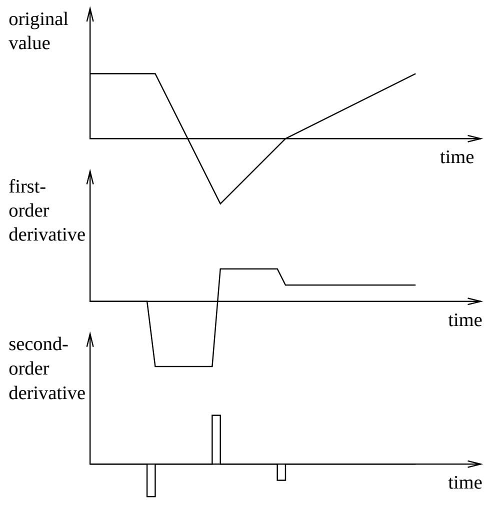

Figure 1. Abstract demonstration of bends in the original graph and its second-order derivative: In places where the original graph shows a bend, the first-order derivative's changes, which causes a peak in the second-order derivative.

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 an observer 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.

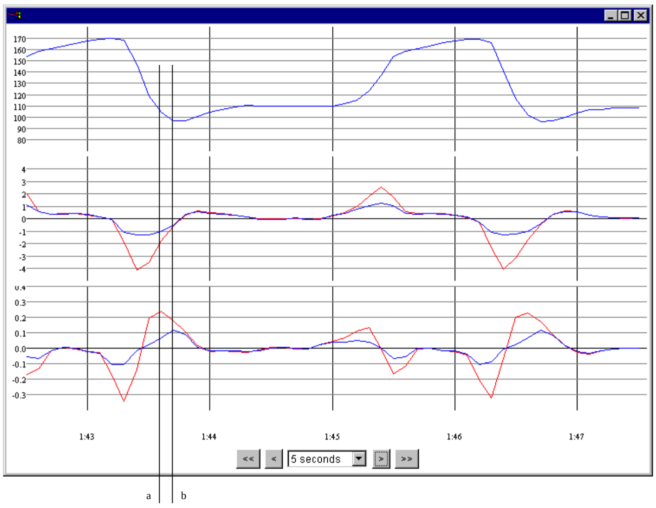

Figure 2. The topmost graph shows the original data. In the middle the gray graph (that with the bigger extrema) shows the absolute value of the first derivative and the black graph shows the angle of inclination. At the bottom, the derivatives of both graphs in the middle are shown. (The more moderate, 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.

Using 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 an indicator of 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, reflecting the indecision of the observer.

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 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

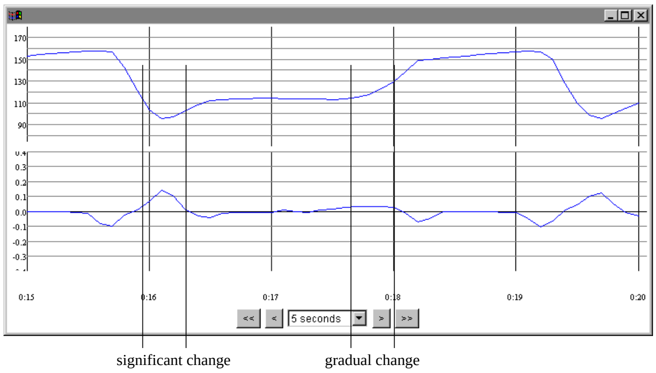

Figure 3. At gradual turns of the original curve (at the top), were a human observer has difficulties in pointing at the exact position of a single corner, the indication function (below) is trapezoidal reflecting the her indecision.

portion of noise in the result of

$$\begin{aligned} \textit{calculated derivative} &= \frac{n * \textit{real derivative} + \textit{noise}}{n} \\ &= \textit{real derivative} + \frac{\textit{noise}}{n} \end{aligned}$$

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.

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 [17], 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

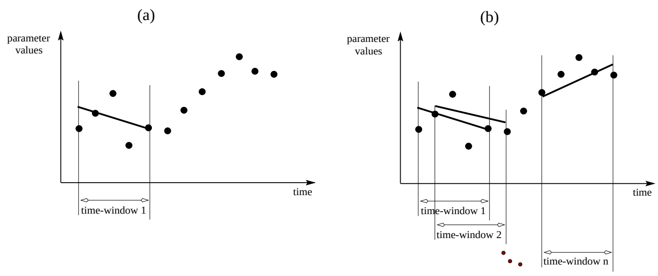

Figure 4. The calculation of the linear regression is done for a time window of fixed size sliding over the entire curve in small steps. (a) shows a single time window and the line calculated from the data points within it. (b) shows a sequence of overlapping time windows and the resulting lines.

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.

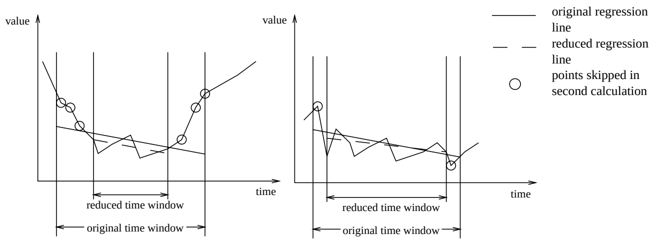

Figure 5. If the graph shows a bend in the interval under consideration (example on the left-hand side), there is a considerable number of data points on each end of the regression line which lie on the same side. Skipping them in the recalculation of the regression reduces the standard error to which 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.

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 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 Matching 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 merged 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 smaller this gain becomes.

3.5 The Resulting Representation

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

3.5.1 Bends

By the term *bend* we subsume the abstract aspects of a turn in the original graph. 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.

3.5.2 Corners

By the term *corner* we describe the position in which the lines neighboring a bend would meet. 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 the influence of noise, in most cases it is necessary to take the average of some of its neighbors into account too. Integrating too many of them in the calculation will distort the result towards the inner side of the bend. Thus, this parameter needs careful optimization.

3.5.3 The Lines in Between

The *lines* between the bends represent the data points of the original graph between two neighboring bends. They 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.

3.6 Relating the Cycles in Oscillations

In many cases, looking at only one oscillation alone is not sufficient, but tendencies developing through a possibly long series of oscillations as well as deviations from the standard or average are interesting. To arrive at this, the following steps are performed:

1. Corresponding bends are found.
2. Their position relative to the start of the oscillation is calculated.
3. The data calculated in step 2 are related to the average of the previous instances of the oscillation.
4. Optionally, slope and standard deviation within a sliding time-window [17] is calculated for any of the values created in step 2 and 3.
5. Optionally, values calculated in steps 2 to 4 can be transformed into qualitative i.e. symbolic representation.

3.6.1 Relating Cycles within an Oscillation

First, for each cycle, a reference point must be found. This should be a point which is known to be invariant itself. A well-tried method is to take a position where the value changes rapidly and steadily, forming a steep slope. In the middle of this section of the graph, its value is chosen as a threshold. The point, where the graph of each oscillation intersects with the horizontal line of the threshold is the reference point of that oscillation. This way, the reference point is most invariant, even in oscillations with varying cycles.

To properly assign corners to groups, just looking at their ordinal number in the stream of corners constituted by each oscillation is not enough. Too often a corner is missing, because it was too small to exceed the threshold, or it is doubled due to errors in the data or the underlying data is simply irregular at that position.

Looking for nearest neighbors suffices only for curves, in which the positions of the same instance of a corner (e.g. the first one) do not change over time. If they do, it is useful to extrapolate the expected position of the next instance of a corner from the previous ones. For this purpose, both the x– and the y–coordinate are considered separately and a linear regression is calculated for each of them.

Often one coordinate (x or y) of a corner is much more regular than the other. In this case, weighting the deviations when looking for near neighbors (with or without extrapolation) helps to improve the result. The more regular coordinate is multiplied by a bigger weight than the other.

3.6.2 Numerical 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 significant the change of the original graph is.
- The x–coordinate shows the position of the corner on the time–axis.

For each line, its inclination is computed.

3.6.3 Numerical 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.

3.6.4 Qualitative Values

The quantities computed in the two steps above can be qualified using a set of tables. For each parameter, a table lists all qualitative values it can take and the numerical limits in between.

4 Fields of Application

In the following, we give some examples of how the data obtained in the previous section can be used.

4.1 Interfacing Knowledge-Based Systems

To bridge the gap between data analysis and knowledge-based systems (KBS) [7], we transform the output into clauses compatible with those use by a KBS.

4.1.1 Symbolic Representation of Features

The values describing bends, corners or lines can be expressed in a list to make them accessible to symbolic reasoners like knowledge-based systems or machine-learning tools.

To improve readability of the output, each corner and each line can be assigned a symbolic name (e.g. P, Q, R, S and T in an ECG) instead of its ordinal number to denote it in an intuitive fashion.

The following example describes a graph consisting of a line increasing by 20 degrees for 100 seconds followed by a narrow bend to the right and 30 seconds of decrease in Clips-Syntax [1]. In this example we omit the corners for clarity.

```
(graph-features(line (100sec up 20))
               (bow (right narrow))
               (line (30sec down 30)))
```

4.1.2 Applying Rules to Detect Patterns in Monitored Data

In the following, we show how knowledge about the interpretation of ECG [8] can be translated into rules and how the data abstracted as described before can be matched with such rules.

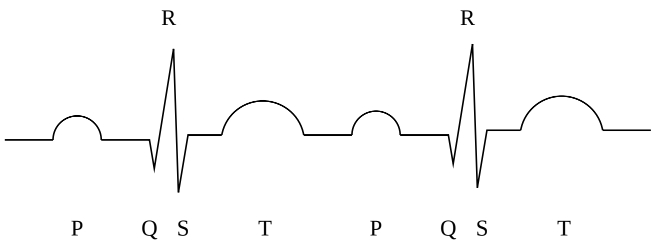

Figure 6. Idealized ECG. The features are labeled by letters. They are explained in Table 1.

Feature	Causality
P wave	excitation of the atria
QRS complex	excitation of the ventricles
$\overline{RR}$ distance	instantaneous heart rate
$\overline{PQ}$ distance	excitation conduction from the atria to the ventricles
$\overline{QS}$ distance	excitation of the ventricles prolonged in case of impeded conduction (heart block)

Table 1. Relation between features of the ECG and underlying causalities.

Figure 6 shows an idealized ECG. Table 1 compares some aspects of an ECG with underlying aspects of the heart and its possible problems. Figure 7 shows some rules in CLIPS [1]

```
(defrule ventricular-conductivity
   (distance (from Q)
             (to S)
             (value ?value):&(> ?value 55))
   (patient  (age ?age):&(and (< ?age 2)
                              (>= ?age 1))))
=>
   (assert (diagnosis (ventricular-conductivity
                       bundle-branch-block)))
```

Figure 7. CLIPS-rules applied on the symbolic representation of an ECG. Note that we measure the distance from peak Q to peak S and not the duration of the QRS-complex as a whole, which is approximately 15 msec longer.

to detect bundle branch blocks and AV-blocks on a symbolic level.

A more detailed analysis of ECG pathology in this setting remains speculative – except for the detection of (ventricular) extrasystoles – because the monitoring ECG is usually not derived under standardized conditions.

4.2 Visualization

The abstraction methods described above produce both qualitative and quantitative information, both on the level of single bends and as attributes of a cycle within the oscillation. In addition to these two dimensions, some features described are relatively rare, e.g., abnormalities in an ECG, others form a steady stream of data, calculated for every instant in time during measurement, e.g. the heart rate.

Sparse events and qualitative information tend to be visualized symbolically, e.g., as bars or markers, while qualitative information is commonly displayed as graphs. For a deeper discussion of visualization aspects see [20, 5].

In the following, we give some examples, namely the display of bend as bars, plotting features of an oscillation over a relatively long period of time and using markers to represent qualitative information.

4.2.1 Displaying Bends as Bars

Bends are features located at a certain position along the time axis and have several abstract attribute, the most important being sharpness and significance. These are derived from the height of the peak in the indication function and the area delimited by the graph indication function and the time axis.

Thus, it seems intuitive to visualize each bend with a significance above a certain threshold as a bar which is equal in height and area to the peak in the indication function.

Figure 8 illustrates this approach applied on data from ergonomic studies in rowing.

4.2.2 Displaying Relative Positions as Graphs

For each corner in the original curve, its position relative to the start of the oscillation or its distance to another corner in the same oscillation can be displayed as a graph.

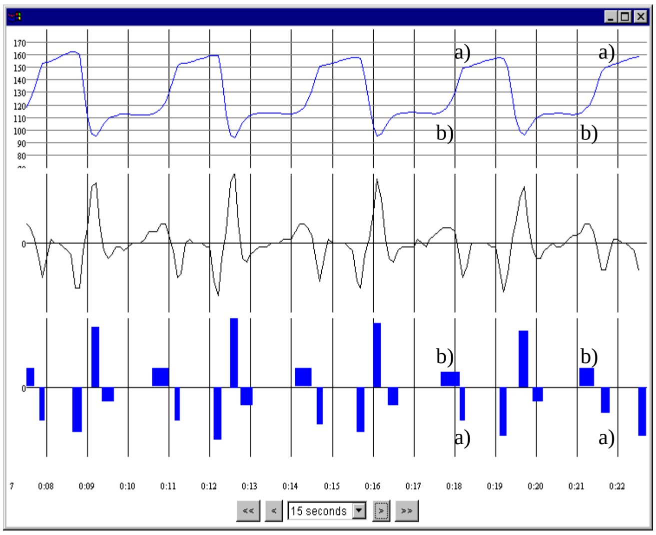

Figure 8. Starting at the top, we show the original graph, the indication function and the bars representing the significant bends. a) shows an example of an irregularity in the curve which a human could also dedect if concentrating on every detail: the bend in the right-most oscillation is not as sharp 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.

Figure 9 gives an example from ECG monitoring. For demonstration purposes, we display the heart rate and the $\overline{QS}$ distance below the very condensed ECG graph.

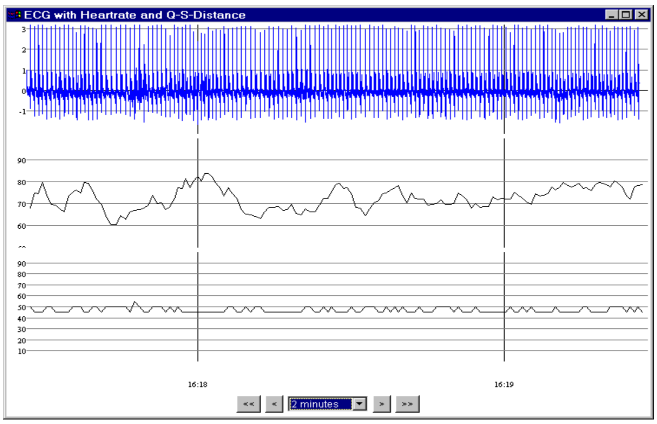

Figure 9. Two minutes ECG sample from an eight hours polygraphic recording of a two years old child. The first graph shows the condensed ECG. The second graph gives the instantaneous heart rate derived from the $\overline{RR}$ distance. The third graph gives the instantaneous $\overline{QS}$ distance. The apparent variability is caused by digitization inaccuracies (resolution is 200 Hz).

4.2.3 Displaying Qualitative Information along the Time Axis

Both the qualitative information derived from lookup-tables as described in section 3.6.4 as well as the results from knowledge-based reasoning as described in section 4.1.2 can be visualized graphically along the time axis.

For rare deviations in the data hinting at significant events, markers in different colors and shape, which are lined up below the original data, are most appropriate.

For continuous information, like qualitative status information, coloring a stripe below according to the values (e.g., blue for low, red for increased, green for normal) yields a dense and intuitive information representation.

Another way to visualize state information is the use of a set of symbols in different colors to represent two or three dimensions in one place. Such a technique has been successfully employed in the VIE-VENT-system [16] for instantaneous status information.

5 Conclusion

We have presented several methods to capture complex rhythmical curves by transforming them into series of bends, corners, and lines, based on the observation that a bend in the curve is synonym to a change in its inclination.

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

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 in addition to the original graph allows for easy detection of trends and outliers which otherwise would be buried in the overwhelming impression of countless oscillations.

Bridge to Knowledge Representation. The abstracted characteristics extracted by our algorithm can be matched against conditions in 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 symbolic machine learning, knowledge-based systems for intelligent alarming and a guideline execution system like as developed in the Asgaard project.

The algorithms described have been implemented in Java^{TM} in an experimental setting to allow their evaluation. Future work will be devoted to the acquisition of rules for the automatic interpretation of clinical data and in the implementation of several modes of graphical display to meet the practical requirements under various settings.

Acknowledgments

We thank Michael Urschitz and Klaus Hammermüller for supplying data and Maria Dobner, Georg Duftschmid, Werner Horn, and Robert Kosara for their useful comments. This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung - FWF" (Austrian Science Funds), P12797-INF.

REFERENCES

[1] 'Clips reference manual', Technical report, Software Technology Branch, Lyndon B. Johnson Space Center, NASA, (1993).
[2] 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).
[3] R. Bellazzi, C. Larizza, P. Magni, S. Montani, and G. De Nicolao, 'Intelligent analysis of clinical time series by combining structural filtering and temporal abstractions', In Horn et al. [10], pp. 261–270.
[4] Daniel Calvelo, Marie-C. Chambrin, Denis Pomorski, and Pierre Ravaux, 'ICU patient state characterisation using machine learning in a time series framework', In Horn et al. [10], pp. 356–360.
[5] Stuart K. Card, *Readings in Information Visualization: Using Vision to Think*, Series in Interactive Technologies, Academic Press/Morgan Kaufmann, 1999.
[6] 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).
[7] J. Giarratano and G. Riley, *Expert Systems - Principles and Programming*, PWS Publishing Company, Boston, second edn., 1994.
[8] W. G. Guntheroth, *Pediatric Electrocardiography*, W. B. Saunders, Philadelphia-London, 1996.
[9] 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).
[10] Werner Horn, Yuval Shahar, Greger Lindberg, Steen Andreassen, and Jeremy Wyatt, eds. volume 1620 of *Lecture Note in Artificial Intelligence*, Aalborg, Denmark, June 1999. Springer.
[11] Jim Hunter and Neil McIntosh, 'Knowledge-based event detection of complex time series data', In Horn et al. [10], pp. 271–280.
[12] E. T. Keravnou, 'Temporal abstraction of medical data: Deriving periodicity', In Lavcač et al. [14], 61–79.
[13] 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)*, pp. 319–30, Berlin, (1997). Springer.
[14] Nada Lavcač, Elpida Keravnou, and Blaž Zupan, eds., Kluwer Academic Publishers, Boston/Dordrecht/London, 1997.
[15] Nada Lavcač, Igor Kononenko, Elpida Keravnou, Matjaž Kukar, and Blaž Zupan, 'Intelligent data analysis for medical diagnosis: Using machine learning and temporal abstraction', *AI Communications*, **11**(3,4), 191–218, (1998).
[16] 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).
[17] S. Miksch, A. Seyfang, W. Horn, and Popow C., 'Abstracting steady qualitative descriptions over time from noisy, high-frequency data', In Horn et al. [10].
[18] Y. Shahar, 'A framework for knowledge-based temporal abstraction', *Artificial Intelligence*, **90**(1-2), 267–98, (1997).
[19] 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).
[20] Edward R. Tufte, *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut, 1983.