

Visualization in the ECG QRS Detection Algorithms

A. Ristovski*, A. Guseva*, M. Gusev**, S. Ristov**

*Innovation Doel, Skopje, Macedonia

**Ss. Cyril and Methodius University, Skopje, Macedonia
info@innovation.com.mk

Abstract—Digital ECG data analysis is a trending concept in the field where applied computer science and medicine coincide. Therefore, in order to meet the requirements that arise, our R&D team has created an environment where developers can test different approaches in data processing. To assist the objective, the platform offers a number of features with the following main target goals: 1) to increase the effectiveness in conducting a proper medical diagnose, 2) to incorporate a unified format of storing the results of the diagnosis conducted and 3) to test various ECG QRS detection algorithms.

Index Terms—ECG Analysis; Modular Software Architecture; AI Agent.

I. INTRODUCTION

The striding process of data digitalization has not left the ECG data immune to it. For an instance, modern professional ECG instruments often feature a digital output in addition to the hard copied records. While hard copied records have the role of feasible hand-outs and provide medical personnel an immediate overview of a patient's condition, the digitalized records have proven to be convenient for storage. Also, there are devices know as Holter monitors, that take continuous readings for a period of one or several days. Holter monitors assist in the diagnosis of conditions that require an extended keep up with the heart's physiology. Their output is inevitably digitalized since it is impossible to keep the lengthly recording on paper. Not long ago, the medical gimmick market became abundant with gadgets that administer the user with simplified ECG readings, again, taking advantage of the most convenient storage form - digitalized data.

No matter the level of exhaustiveness and accuracy of the information contained in the digitalized data, many opportunities rise in the filed of digital data processing. What is of greatest significance, is the opportunity to give a diagnosis on the recordings without the assistance of medical personnel, i.e. to replace human resources with a virtual Artificial Intelligence (AI) agent [1].

The extent of details provided in a conducted diagnosis vary according to the needs. In some cases, only statistical data targeting a specific medical condition or property is collected, and in others, a thorough analysis of the ECG recording is handled - to deliver an accurate cardiac profiling.

Regardless the scope of the diagnosis, visualizing the outcome of the signal processing while testing different algorithms is of substantial benefit. Applying a filter to the signal

before it is processed can increase the effectiveness of the algorithm used for the processing. Therefore, for the purpose of having an insight how the filter has effected the initial signal, a graphic representation of the processed signal in contrast to a representation of the original one is a must. That way, by modifying the filters' parameters and immediate observation of the results, the process of improving the effectiveness gains a significant advance.

An ECG signal consists of a sequence of form-specific element set of typical waves that appear in a unambiguous pattern, at least to say unambiguous for most of time. These typical waves are referred to as components of the signal or characteristic points. Although the signal's components can be easily distinguished by the human eye, it is quite the challenge to teach an AI agent how to do so. By visually marking which signal components have been detected and their exact point of detection, one can determine to what extent the detection algorithm has been successful. That way, the left-outs can be consequently analyzed and the algorithm can be modified accordingly, hence, doing corrections in a favorable direction.

Some algorithms may even use pattern recognition in the detection of the signals' components. By visualizing how does a component fit in the pattern it is supposed to fit in, the pattern can be improved in a way so that it better detects the components that have been left out during the detection. In addition, one of the more advanced features when using a pattern recognition is the ability to asses which part of the components deviates from the general form and in what way.

Therefore, using visualization in the development of algorithms which do the ECG component detection is of great interest. The goal is to design an AI agent that will mimic the way a medic would interpret a reading. Although the AI agent itself does not need visualization of any kind to do its job, it is practically inevitable for the development process to feature the use of visualization modules, i.e. provide developers a visual insight in how the approach they are using is affecting the correlation between the input data - the ECG signal and the output data - a legitimate diagnosis. Whereas the paper is focused on the role of visualization in the AI agent design process, it also encloses an architectural design of an ECG diagnosis application that is not only aiming to deliver accurate results, but also, to incorporate functionalities for testing new approaches and improve overall effectiveness.

The paper is organized as follows. The architectural design of the solution is given in Section II. Section III describes the architectural modules in detail and explains how the visualization has been implemented how it assists the process of devising the detection algorithms. An overview of existing related work is handed in Section IV. Section V discusses the conclusions and directions for future work.

II. ARCHITECTURAL DESIGN

The features of a tool designed for delivering an ECG diagnosis are incorporated into the modules of a system with modular software architecture. Such design allows the advantages of distributed development, code re-usability and modular transparency, hence, the integration of new functionalities and the upgrade of existing ones can be done easily.

The process of AI agent assisted diagnosing is very close to how a medic would act upon doing so, with respect to the steps needed to utilize the input and output formats that make a notable exception. The whole routine consist of several stages. These include:

- Unpacking and interpreting the ECG recording, whatever the file format;
- Pinpointing the signal components of the signal:
 - P waves;
 - QRS complexes;
 - T waves;
- Analyse the constellation and form of the signal components;
- Set a medical diagnosis;
- Save the annotations;

In addition, the system is capable to evaluate the effectiveness of the applied algorithm on an extensive database of patients. It responds with several key effectiveness indicators, such as *detection rate* (percentage of detected features), *hit rate* (percentage of correctly detected features), *miss rate* (percentage of incorrectly detected features), and also the *extra rate* (percentage of extra determined features, which are not recognized by a human and have to be eliminated). This module is efficient in recognizing those parts of the ECG which caused problems (extra features or misses) and a user can automatically analyze them and improve the algorithm by customizing and adapting its parameters.

The product reviewed in this paper has been designed in accordance with the stages of ECG analysis numbered above and is using a separate software module for each stage. Guidelines for pragmatic processing and several assisting visualization modules are also designed in accordance with the concept of modular architecture. They have been offered at disposal in order to make the whole process simpler and more transparent for developers. Although the architecture makes a notably pragmatic solution for the problem, what is more important is that it excels at increasing the effectiveness of the diagnostic capabilities.

The overall architecture of the system is presented in Fig. 1. The next section gives a description on how each module deals

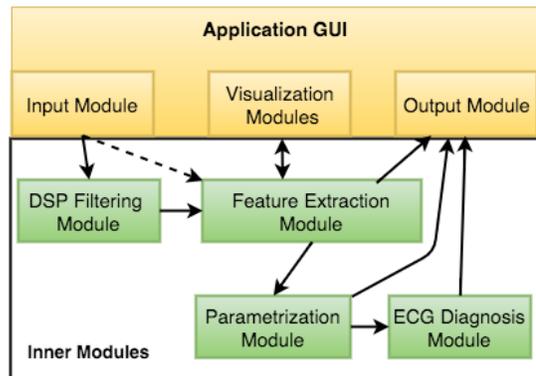


Fig. 1. An overview of the system's architecture

with the problems in respect, and, moreover, focuses on the interest of implementing the modules for visualization.

III. FUNCTIONAL DESCRIPTION

One of the fundamental ideas for an application with such architecture is interoperability. That is achieved by allowing data input in different file formats, hence, making it possible for the application to process output data from a variety of ECG sensors. The application's *input module* handles the different data representations containing the actual ECG signal, so that the data is given a unified composition - an inevitable prerequisite for the next stage of the analysis.

There are several standard medical record data formats for storing and retrieving ECG files. Although most of them are XLM based, still there are examples of binary files, such as those used by PhysioNet open source library [2]. The most profound is maybe the USA FDA standard HL7 [3] that specifies annotated ECGs and provides means to systematically evaluate the ECG waveforms and measurement locations. Bond et al. [4] review nine different formats used to store the ECG, such as SCP-ECG, DICOM-ECG, and HL7 aECG. The input module also does the consolidation of additional record information, such as sensor ID, time stamp, location, type of ECG lead being analyzed [5] and alike.

The *ECG signal visualization module* incorporated into the application's GUI is an important module for visualization of the ECG signal. By the means of this feature, the developers have an immediate insight on how does the analog representation of the digital signal look like, as if looking at a hard copied ECG recording.

Therefore, this visualization module is responsible for displaying the consolidated ECG signal. The OpenGL libraries [6] have been used for the purpose of visualizing the data. Since the development environment has incorporated the C# programming language, a wrapper for the OpenGL libraries is needed. The choice of OpenGL wrapper has come down to the OpenTK wrapper [7], superseding the Tao Framework [8].

The application interface displays the additional information of the ECG signal and highlights the analog representation of the ECG signal in a standard ECG graph paper surrounding [9]. A standard record duration of 10 and 30 seconds are the

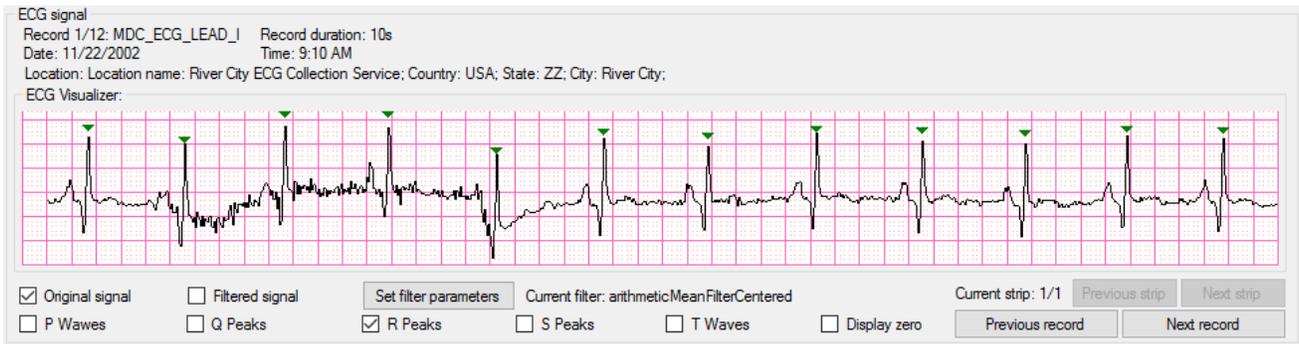


Fig. 2. A snapshot of the signal visualization GUI component

most common length of an ECG record, as the information contained within a 10 seconds reading is sufficient for a standard diagnosis. Moreover, records with these lengths are optimal in terms of appearance since longer records would make the visualized signal appear cluttered.

However, records with longer duration are not nonexistent. Certain diagnosis types require readings that last up to half an hour. Visualization of such records has been also supported, by enabling transition between consecutive 10 second strips, thus, maintaining optimal appearance.

The size of the visualization component in the Graphical User Interface (GUI) is scalable, so that it adapts to the dimensions it is supposed to have, hence, the ECG recording being displayed scales up to provide maximum clarity and visibility. The height of the graph paper squares scales up as well, to conform with the signal's amplitude. Optionally, a zero amplitude line can be displayed as well.

The *DSP module* is another auxiliary application module used for filtering the digital signal from noise. The effectiveness of the upcoming second phase of ECG analysis can be significantly improved by applying a filter to the signal. The filtering can realize various DSP methods, such as, interpolation, moving average, regressive filters, differential filtering, wavelets, discrete Fourier transformations, etc. The module allows use of a combination of any of these filters and customizing their parameters.

The processed signal is at request displayed within the same visualization module, next to the original signal - in order to give the developer an insight of how applying the filter and changing the filters' parameters is affecting the signal. The GUI component is shown on Fig. 2. The way the ECG signal visualization module and the DSP module improve the effectiveness is shown on Fig. 3.

The second phase involves *the feature extraction module* used for detection and localization of the signal components. The easiest way of doing this is to conduct a QRS complex detection first, as this signal component has the most distinctive form and is the easiest to pick up out of the ECG recording. The other two components, the P waves and T waves, can be optionally detected as well - by conducting a look up for one of each in the interval between two neighbor QRS

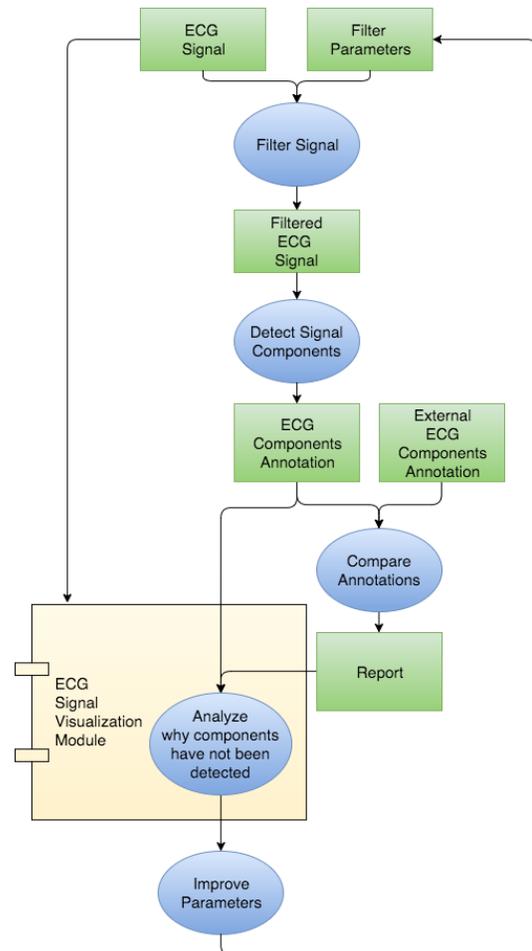


Fig. 3. An iterative process on how the ECG signal visualization improves the overall effectiveness

complexes. This stage can be done in a number of different ways: by analyzing the signal's curve slopes, by using pattern recognition, by looking for a local maximum at predefined intervals, etc. It is up to the developer to make an assessment which approach is of greatest interest. All the detected signal components are then collected in a logical structure, referred to as ECG signal component annotations, where an annotation

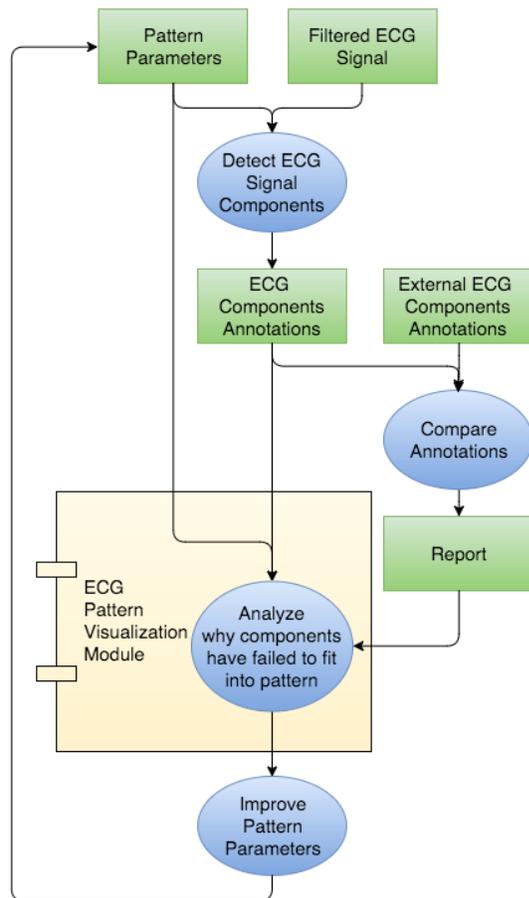


Fig. 4. An iterative process on how the pattern visualization model improves the overall effectiveness

consist of the following couple: component index (time stamp) and component amplitude.

In case of using a pattern recognition, the feature extraction module displays the pattern and the signal component entering the pattern. Because there are multiple components detected, this module enables selection of the signal component displayed on the ECG pattern visualization module. In case of undetected component, a specific part of the signal can be put on display under the pattern so that the pattern can be modified in order to better detect the components. This process is explained on Fig. 4.

In case of using the expert system to diagnose a streaming data from an ECG sensor as a web service, the feature extraction module offers an excellent opportunity to tune and customize various filter and feature extraction parameters of the algorithms to match the specifics of the person.

The ECG signal visualization module can also help with testing the effectiveness of stage two by specifically marking the signal components, i.e. by indicating which components have been successfully detected and which have not. The application's interface offers selection of the signal components that are being displayed. For each type of components, if specified for display, an indicator is drawn right above their positions.

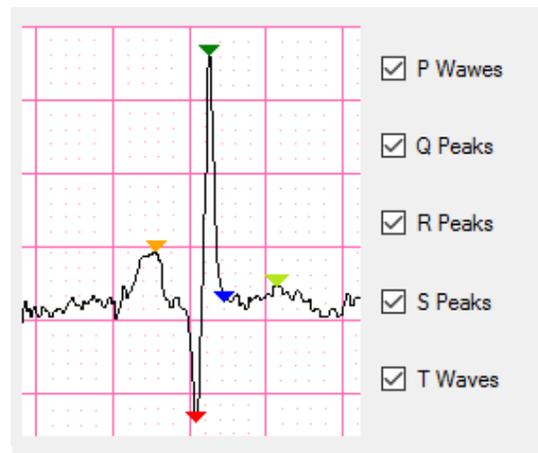


Fig. 5. A snapshot of the ECG component detection visualization

The feature is shown on Figure 5.

After the signal's components have been temporally pinpointed (localized), the third phase is realized by the *parametrization module*. It features two types of analysis: obtaining temporal parameters and obtaining vector polarization parameters. The temporal parameters' analysis starts with the calculation of typical component intervals. These intervals optionally include: PR, QRS, QT, PP and RR interval duration. Out of that data, additional temporal components are derived, such as Beats Per Minute (BPM), Heart Rate Variation (HRV), Beat Fluctuation (BFx), SDNN and other related parameters [10]. Then, the form of the signal components is being investigated into for the vector polarization analysis. Although there is small number of options how the parametrization can be done, since it comes down to a standard medical approach, the way the component extraction is handled offers significant flexibility and is, in fact, very similar to how the second phase - the component detection, is accomplished, as the signal's amplitude is taken into consideration. In case of having only one lead in the ECG recording, very little relevant information can be given for the vector polarization, because the vector polarization is a derived compound from several leads that make up the QRS phasor [5]. Therefore, when processing one lead record, the vector polarization comprises only parameters such as abnormal amplitude levels and component width.

The fourth phase consists of the *diagnosis module* that can closely correlate to how a cardiologist would conduct the ECG analysis. The information collected at the third phase is being used - a combination of a number of parameters, would trigger a medical condition marker. All the triggered markers are then reasoned into a closing composite diagnosis.

Eventually, all the information gathered at the previous phases: the ECG signal and the additional ECG record data (phase one), the ECG component annotations (phase two), the typical ECG parameters (phase three) and the ECG diagnosis (phase four) are consolidated within the *output module* into an output file. Once again, it is up to the developer to determine which information is of interest and needs to be included in

the output. The output data can then be stored into a data base, sent through a network channel or stored locally as an individual file.

Additional functionality of the final phase is the effectiveness check. There are databases that in addition to the ECG recordings offer supplemental component annotations. The annotations generated by the developed system and the pre-generated annotations supplemented to the ECG recordings from the databases can be automatically compared, via another auxiliary module, inbuilt in the output module. An extensive report of such comparison optionally accompanies the output file, and it includes analysis of the effectiveness indicators, such as the detection rate, hit rate, miss rate and extra rate.

The application has a fully functional Graphical User Interface (GUI) that allows the developer to have an actual overview of the diagnosis given, as well as interfaces for handling the application's input and output, a debug window and a status pane, as shown on Figure 6. There are additional interfaces for customizing the parameters used during the component detection process and for customizing the parameters used for the signal filtering.

IV. RELATED WORK

There are several tools that help ECG diagnosis, and most of them are open source software packages or offered as web application. ECG has been attracting the researchers a lot and even the mathematical toolkits Mathematica [11] and MatLab [12] offer extensions for ECG interpretation.

PhysioNet [13] have developed a large collection of software for viewing, analyzing, and creating recordings of physiologic signals, and their product Wave is a visual tool that enables an extensible interactive graphical environment for manipulating sets of digitized signals with optional annotations.

An ECG simulator is a device, which can simulate ECG signals recorded previously in the real time mode [14]. The main goal of ECG simulators is to convert the digital ECG signals to analogue counterparts, which have been created in a computer graphic environment [15]. The ECG Simulator enables analysis and study of normal and abnormal ECG waveforms without actually using the ECG machine. One can simulate any given ECG waveform using the ECG Simulator [16]. This simulator can be used both for clinical training of doctors as well as for design, development and testing of automatic ECG machine without having real subject for this purpose.

Most of the simulators on the market are offering services for training of readings of ECG and similar practical clinical skills. Nilsson et al. [17] propose a web-based ECG-interpretation program for undergraduate medical students. Lu et al. [18] designed a tool for modeling and simulation of electrophysiology. Kaur describes a simulator [14] that produces realistic ECG with the possibility of including other biological signals, e.g. blood pressure.

In the literature, software-based ECG simulators have been developed by the help of different programming languages

such as Java, C++, C#, and MATLAB. Oefinger et al. proposed a web service Java and CGI-based, which can plot ECG signals in the ECG database called Physionet [19]. Kartnik designed a MATLAB based ECG simulator [16], which can generate normal lead II ECG wave forms.

Another example of a web-based ECG simulator is the "The Six Second ECG Simulator" [20]. The simulator "simECG" was developed using C++ language [21]. Additionally, an ECG simulator of particular interest is WebECG [22] where various ECG signals can be generated and plotted in 3-dimensional view with zooming and moving.

Therefore, most of the related available products are in fact ECG simulators, meant for medical training of medical personnel or medical software developers. However, the system proposed in this paper would assist the medical software developers in a unique way. That is, not for training individuals on how to read and interpret ECG recordings, but to enable developers to design approaches for digital ECG processing and analysis. On the other hand, the proposed solution has functional similarities to other applications meant for diagnosis, such as the products by PhysioNet.

V. CONCLUSION AND FUTURE WORK

This developers' environment has been built so that it meets the needs of an ongoing project. The project involves analysis of ECG data recorded by a set of wearable ECG sensors. An AI agent serves the sensors and is capable of detecting any abnormalities in the subject's cardiac condition and also, capable of preventing critical scenarios on time.

So far, the application has substantially contributed to improving the effectiveness of the component detection in a number of ways: by improving the pattern recognition approach and by testing different component detection methods, by contributing in the analysis of how different filters affect the original signal, by following a strict component based architecture which increases the code re-usability in case of transitioning to other platforms.

One of the features that this tool offers is the transitioning from a desktop application to a web service. This is of high interest, because that way out R&D team could continue working on a fully web based platform, hence, make the corrections to the web server that is in charge of the processing of incoming ECG records without much effort.

Howbeit, a greater variety of filters can be implemented, so that the users can choose one or combination of several, all in order to increase the accuracy of the diagnostics. At the same time, most of the added features have an effect that can be seen "on the go" - all thanks to the visualizer module.

To conclude, although the software was designed for a specific cause, it can easily be re-purposed or upgraded with functionalities, all due to its architectural modularity. It is a product that is not of exclusive interest to the team that build it, but assists the work of others that deal with similar challenges.

The main benefit of this tool is the possibility to fine tune the parameters for DSP filtering and feature extraction in the ECG detection and, therefore, customize it towards a specific

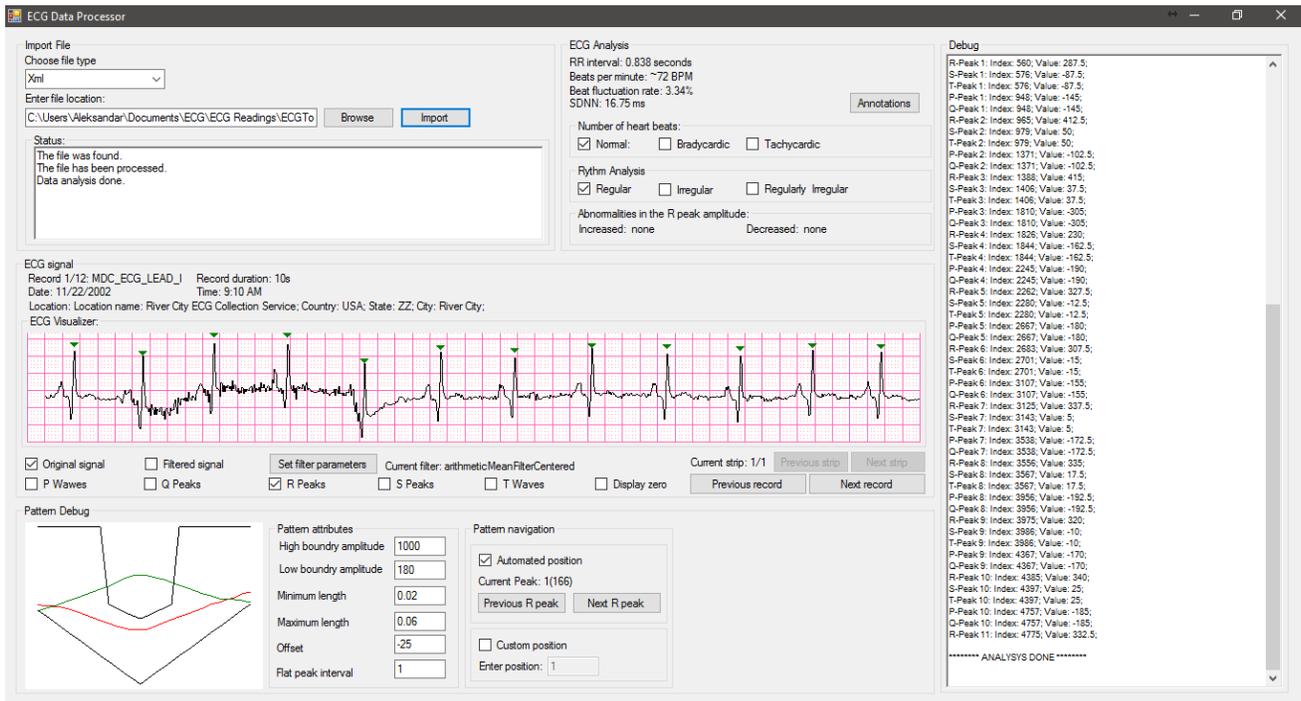


Fig. 6. A snapshot of the visual tool for ECG QRS detection

patient, especially in a case of using a streaming ECG sensor that continuously sends data to a diagnosis system. A machine learning approach can be used for the pattern parametrization in order to boost the effectiveness of the expert system.

As future work, we plan to develop more filters and feature extraction algorithms. Additionally, the system is planned to support simultaneous processing of multiple recordings, hence, it will allow handling of entire databases and record sets. The effectiveness check that covers the annotations generated from a single file, will be extended to cover the annotations of multiple files and subject them to extensive statistical analysis. We believe that such expert system would be of particular interest for students and medical researchers.

REFERENCES

- [1] F. Gritzali, "Towards a generalized scheme for qrs detection in ecg waveforms," *Signal processing*, vol. 15, no. 2, pp. 183–192, 1988.
- [2] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [3] Health Level Seven International. (2015, June) H17 version 3 implementation guide: Annotated ECG (aECG). [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=102
- [4] R. R. Bond, D. D. Finlay, C. D. Nugent, and G. Moore, "A review of ECG storage formats," *International journal of medical informatics*, vol. 80, no. 10, pp. 681–697, 2011.
- [5] T. B. Garcia *et al.*, *12-lead ECG: The art of interpretation*. Jones & Bartlett Publishers, 2013.
- [6] D. Shreiner, M. Woo, J. Neider, T. Davis *et al.*, *OpenGL (R) programming guide: The official guide to learning OpenGL (R), version 2.1*. Addison-Wesley Professional, 2007.
- [7] S. Apostolopoulos, "The Open Toolkit library online documentation," <http://www.opentk.com/doc>.
- [8] D. Hudson, R. Ridge, R. Loach *et al.*, "The Tao Framework," <http://sourceforge.net/p/taoframework/wiki/Home/>.
- [9] E. Einthoven's, "Ecg graph paper."
- [10] M. Malik, "Heart rate variability," *Annals of Noninvasive Electrocardiology*, vol. 1, no. 2, pp. 151–181, 1996.
- [11] T. Pogorelov. (1998, Sep) Ekg qrs detection algorithm. [Online]. Available: <http://library.wolfram.com/infocenter/Demos/4476/>
- [12] R. Gupta, J. Bera, and M. Mitra, "Development of an embedded system and matlab-based gui for online acquisition and analysis of ecg signal," *Measurement*, vol. 43, no. 9, pp. 1119–1126, 2010.
- [13] PhysioNet.org. (2015, July) The WFDB software package, software for viewing, analyzing, and creating recordings of physiologic signals. [Online]. Available: <https://www.physionet.org/physiotools/wfdb.shtml>
- [14] G. Kaur, "Design and development of dual channel ecg simulator and peak detector," *Thapar Institute of Engineering and Technology, Deemed University, Patiala*, 2006.
- [15] C. Caner, M. Engin, and E. Z. Engin, "The programmable ecg simulator," *Journal of Medical Systems*, vol. 32, no. 4, pp. 355–359, 2008.
- [16] R. Karthik. (2003) Ecg simulation using matlab. [Online]. Available: www.mathworks.com/matlabcentral/fileexchange/10858
- [17] M. Nilsson, G. Bolinder, C. Held, B.-L. Johansson, U. Fors, and J. Östergren, "Evaluation of a web-based ecg-interpretation programme for undergraduate medical students," *BMC medical education*, vol. 8, no. 1, p. 25, 2008.
- [18] W. Lu, D. Wei, X. Zhu, and W. Chen, "A computer model based on real anatomy for electrophysiology study," *Advances in Engineering Software*, vol. 42, no. 7, pp. 463–476, 2011.
- [19] M. Oefinger and R. Mark, "A web-based tool for visualization and collaborative annotation of physiological databases," in *Computers in Cardiology, 2005. IEEE, 2005*, pp. 163–165.
- [20] S. L. Canada. (2016) The six second ecg (cardiac rhythm simulator). [Online]. Available: <http://skillstat.com/tools/ecg-simulator>
- [21] M. J. Martins AC, Costa PD. (2016) simecg: Ecg simulator. [Online]. Available: <http://simecg.sourceforge.net/>
- [22] E. Güney, Z. Ekşi, and M. Çakıroğlu, "Webecg: A novel ecg simulator based on matlab web figure," *Advances in Engineering Software*, vol. 45, no. 1, pp. 167–174, 2012.