Health and Technology (2020) 10:417–421
https://doi.org/10.1007/s12553-019-00364-x

ORIGINAL PAPER

Development of a virtual reality application for interactive visualization of respiratory sounds signals

Rodrigo Castillo-Gonzalez[1] · David A. Cerda-Davila[1] · Sonia Charleston-Villalobos[2] · Angel T. Aljama-Corrales[2] · Bersain A. Reyes[1]

Received: 15 January 2019 / Accepted: 22 August 2019 / Published online: 29 August 2019

Abstract
Waveform analysis of Respiratory Sound (RS) signals has been useful to identify and characterize adventitious sounds, both discontinuous (transients and explosive with a duration of about 5 and 10 ms) and continuous (quasiperiodic and musical with a duration of more than 100 ms), related to pulmonary pathologies. Thus, a medical doctor aided with time expansion of RS is able to detect the presence of these events and use information of their counting, location, and characteristics to assist the clinical diagnosis. However, the visual analysis of RS in conventional monitors becomes slow and tedious, which limits its use in health centers because of lack of time, especially in developing countries, where an excessive workload is a common scenario. This work proposes an interactive visualization system as an alternative to address such limitations. The system uses Virtual Reality (VR) to enable an easy and natural manipulation of displayed RS signals within isolation environments.

Keywords Virtual reality · Respiratory sounds (RS) · Signal analysis · Interactivity

1 Introduction

Acoustic information gathered during thoracic auscultation is still of great importance to diagnose diseases and evaluate pulmonary function. However, the random nature of respiratory sounds (RS), as well as limitations of human auditory perception, have resulted in variable and non-standardized nomenclature, and documentation of RS [1].

Computerized Respiratory Sound Analysis (CORSA) has provided quantitative tools to establish parameters and standards that support physicians and researchers in their labors. Studies in the frequency domain have provided information about the nature of ventilatory, abnormal, and adventitious sounds (superimposed to basal sounds). In addition, experts use Time Expanded Waveform Analysis (TEWA) to study short-lasting sounds such as crackles in patients with diffuse interstitial diseases, and wheezes in patients with obstructive pulmonary diseases [2]. Hence, in addition to auscultation, physicians trained and familiar with RS can improve clinical diagnosis by observing RS signals in a conventional monitor. However, given the relative short duration and spectral characteristics of acoustical events of interest (e.g. crackles have a transient behavior lasting around 5 or 10 ms with a time-varying spectrum), the analysis of patient RS registries, even with a duration lasting less than 1 min, is a cumbersome and tedious task. However, the complexity of this analysis increases not only with the duration of the recordings, but also with excessive workload that most health care professionals experience [3]. A similar complex scenario, regarding the visual analysis of RS, can be found in research environments.

✉ Rodrigo Castillo-Gonzalez
rodrigocastillogl@gmail.com

David A. Cerda-Davila
david_alejandrodavila@hotmail.com

Sonia Charleston-Villalobos
schv@xanum.uam.mx

Angel T. Aljama-Corrales
alja@xanum.uam.mx

Bersain A. Reyes
bersain.reyes@uaslp.mx

[1] Biomedical Engineering, Faculty of Sciences, Universidad Autónoma de San Luis Potosí, San Luis Potosí, San Luis Potosí, Mexico

[2] Electrical Engineering Department, Universidad Autónoma Metropolitana, Iztapalapa, Mexico City, Mexico

The handling of large amounts of complex information has motivated the development of new visualization systems that ease analysis, interpretation, and manipulation of such complex information. Among visualization systems, those with immersive characteristics are particularly interesting given their higher interaction capabilities through natural and intuitive interfaces [4]. Accordingly, a Virtual Reality (VR) environment can stimulate the learning and comprehension of information by providing a narrow link between the data-driven information and reality [5]. The benefits of systems for graphical manipulation in VR have been studied in terms of the accuracy, time, and difficulty level by comparing the performance of user subjects to perform tasks, such as graphics displacement, expansions, and rotation in VR applications versus those obtained via a conventional monitor and mouse [6]. Although no statistically significant differences have been found regarding accuracy, time, and difficulty level, users provide a positive evaluation of the VR system in terms of being more intuitive, easy to learn, engaging and interesting.

The goal of this work is to present the development of a computational VR application (app) for RS signals display and interactive manipulation. The application could be used as an alternative analysis tool for physicians and researchers, as it provides an isolated environment that eliminates distractions. Similarly, the application is easy to use. The VR system was developed having in mind the advantages of VR environments not only to display signals, but also to increase physician and researcher involvement in RS signal analysis using a more natural and precise manipulation technique within isolation environments. The VR app was programmed using C# language and makes use of the Oculus Rift system.

2 Methodology

2.1 Virtual reality hardware

We used the VR Oculus Rift (Oculus VR, Facebook Inc., CA, USA) to ensure a VR experience free of blurring and judder effects, commonly found in conventional monitors. As Fig. 1 depicts, the Oculus Rift system comprises the following components:

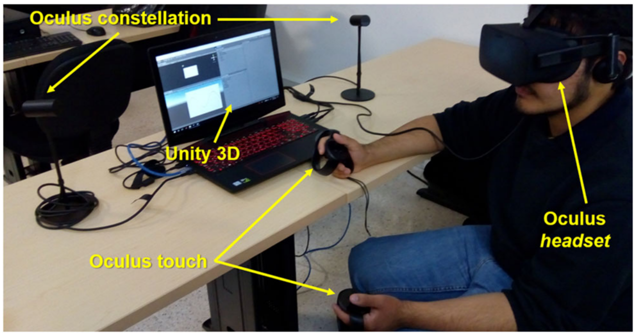

Fig. 1 Virtual Reality Oculus Rift system integrated with IDE Unity 3D

- Oculus headset (VR glasses). Two OLED displays with 1080 × 1200 resolution, 90 Hz global refresh rate, low image persistence, 110° field of view, and 3D audio.
- Oculus constellation (positional tracking system). Infrared (IR) LEDs for precise tracking of VR glasses and other VR devices (e.g. controls, with a sub-millimeter accuracy and low-latency).
- Oculus touch (motion controllers). Two manual controllers with an analog joystick, three buttons and two shutters employed to focus or grab. Motion controllers can detect finger gestures performed by the user.

2.2 Virtual reality developing software

To develop the VR app, we employed software Unity 3D (Unity Technologies, CA, USA), which is highly accessible and has a large amount of documentation. Unity is a cross-platform game engine used to develop 3D video games. The engine uses an Integrated Development Environment (IDE) and can add scripts developed in JavaScript, C#, or Python Boo. In addition to game development, Unity can be used to develop interactive content in VR [7]. In this sense, the engine allows developers to integrate the VR Oculus Rift system, as well as the official Oculus Integration package, which offers multiple resources to incorporate VR capabilities to any project. We employed the VRTK (VR Toolkit) package to govern the interaction with the camera, and to use the controllers to manipulate the graphical user interface.

2.3 Computational system

To comply with the computational resources needed to create and manage a VR environment on Unity, we used a Lenovo™ Legion Y720 computer with Intel Core I7-7700HQ processor, 16 GB of DDR4 RAM, NVIDIA GeForce GTX1060 graphical card with 6 GB, and USB 3.0 ports. Such technical characteristics also satisfy the minimum requirements of the VR Oculus Rift system.

2.4 Acquisition of physiological signals

To develop the VR app, we acquired signals from a healthy volunteer subject without cardiorespiratory diseases after obtaining his informed consent, as suggested by the Declaration of Helsinki. In particular, we registered tracheal RS via an acoustical sensor composed of a subminiature electret microphone (Knowles Electronics, IL, USA) encased in a plastic chamber, specifically designed for RS acquisition. Simultaneously, we recorded the respiratory airflow signal and the cardiac electrical activity via a Fleisch-type pneumotachometer and an electrocardiograph, respectively.

We obtained all signals using an A/D converter with sampling rate of 10 kHz and 16 bits-per-sample. The respiratory maneuver lasted about 15 s and consisted of an initial period of apnea (~ 2 s), 10 s of respiration at a maximum airflow of 1.5 L/s, and a final apnea period (~ 2 s). The acquired signals were stored in a comma-separated text file for their further loading and processing in the VR app.

2.5 Development of the VR app

Initially, we used Unity native objects to connect the signal amplitude values at each time instant via linear interpolation. In 3D, such native objects could be thin cylinders, which allows them to remain visible without being distorted or out of place. We could trace 1D lines with a width factor given a set of space coordinates. Figure 2 depicts the graph of the RS, airflow, and ECG waveforms generated by this first approach. However, we experienced some issues using these native objects: 1) the image of a floating graph in a 3D space is confusing and does not easily provide a reference about its location, and 2) the large number of generated cylinders were of great computational burden.

To overcome the abovementioned limitations, we used a Canvas object, consisting of a space to develop a Graphical User Interface (GUI). The Canvas area is displayed as a rectangle in the scene view, in a screen fashion. It is possible to select elements inside this area, and the waveforms of RS and other signals were displayed inside the area. Consequently, their display has now a reference location and by having a platform on which to draw the graph, the signals do not appear floating on an empty space.

Even though Unity has elements to draw line segments, these could not be directly integrated into the Canvas space. Hence, we added the Unity UI Extension package to the project to incorporate diverse elements to the GUIs, including the object UI line, which draws a line between two given points. Accordingly, we defined the position and dimensions of the visualization area, which in turn established a spatial reference frame. We generated points in the format of (time, amplitude) coordinates, such that for two consecutive points a UI line object was drawn by following the line equation:

$$\frac{y-\mathrm{Y}_i}{x-\mathrm{X}_i}=\frac{\mathrm{Y}_{i+1}-\mathrm{Y}_i}{\mathrm{X}_{i+1}-\mathrm{X}_i} \tag{1}$$

where (x,y) corresponds to the points coordinates on the line that connects the i-th point (X,Y) with the next one.

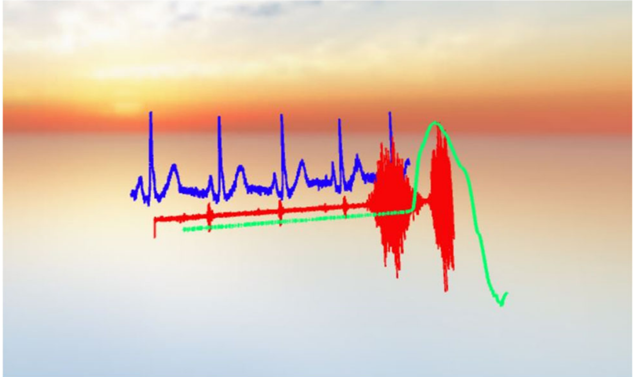

Fig. 2 Waveforms displayed in Unity by means of native objects. Respiratory sounds (red), airflow (green) y ECG (blue)

In the script of the app we loaded a text file containing the signals to display in the VR environment. It is worth mentioning that the sampling frequency of 10 kHz satisfies the Nyquist criteria for respiratory sounds, which have spectral components of up to more than 3 kHz. However, such sampling frequency leads to too many graphical objects in Unity (15 s × 10 kHz × 3 channels = 450,000 points) that slow the performance of our application. Hence, we generated new text files of the acquired signals inside the VR app by downsampling them at lower frequencies, so that they are employed at upper visualization scales (i.e., to visualize the signals at durations close to the full duration), due that at these larger scales the details are not necessary and are even unappreciated. As the signals are expanded in time (i.e., the time scale is reduced), the number of samples in the visualization cannot be reduced because the fidelity of the signal results essential for the analysis via TEWA technique. Hence, segments of the signals at higher sampling frequencies should be employed. Accordingly, the VR app alternates between the different text file versions in response to the time expansion factor selected by the user.

We created location coordinates of each sample of the signal using a vector X for each time instant and a vector Y for the corresponding amplitude of the signal. We normalized amplitude vector Y by the maximum value found in its rectified version (absolute value) and the dimensions of the Canvas space. Next, we multiplied the normalized vector Y by a gain factor initialized at 1. We adjusted vector X so that, initially, the waveforms were displayed at their total duration; then, we divided vector X by a time expansion factor, also initialized at 1.

At this stage, it was just necessary to add a tool to modify time expansion factor and amplitude gain factor to perform zoom-in and zoom-out functionalities. Thus, we added a horizontal sliding bar to modify and actualize the time expansion factor. Similarly, we added a vertical sliding bar to modify the amplitude gain factor. We displayed both bars in the Canvas space. In addition, we added a horizontal scroll bar and a vertical one to perform displacement along the signal extension. We used buttons of the Oculus touch to control these bars. By pressing the analog joystick, a pointing control is activated, and the selection on the GUI could be done via the rear shutter.

3 Results

Figure 3 depicts the graph generated by the VR app. Only RS (red) and airflow (green) signals were selected to be displayed. This representation is called phononeumogram and let us know the temporal occurrence of acoustical events. Initially, the app displays the full-time waveforms, which in this example correspond to 15 s. In this initial display range, the initial apnea, respiratory sounds (inspiratory and expiratory), and final apnea are visible. The horizontal bar on the top-right corner of the white space, shown in Fig. 3, is used to modify the expansion factor in time, while the vertical one is used to modify the gain factor in amplitude.

The horizontal bar next to the title of the graph corresponds to the general gain factor applied to the signal. This horizontal bar automatically modifies both expansion in time and amplitude. Figure 4 depicts a graph of the same previous, but over an interval of 9.37 to 9.56 s. Signals are zoomed by pointing and moving the control of the horizontal bar via manipulation of the Oculus touch control. When the amplitude of the signal and the expansion over time are modified, we observe high frequency signal details (low time scales), as expected. In these cases, the user uses the scroll bars deployed at the base and side of the screen, to scroll along the signals in time and amplitude.

4 Discussion

This paper presents the development of a VR application that can help physicians identify events of interest in RS signals. The application was developed in Unity 3D by implementing basic functionalities for signal interactive manipulation, such as data loading, signal selection, resampling, display, scaling, and displacement in both time and amplitude, for the VR Oculus Rift system. We managed to plot and interactively manipulate the signals with the application. Due to the dynamic selection of signals at different sampling frequencies, our VR app does not present problems with the available computational resources. As main benefits, our VR system increases direct interaction between users and the obtained signals and provides an isolated environment.

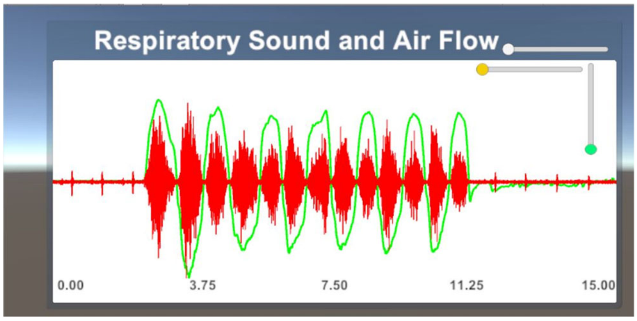

Fig. 3 Interactive display of respiratory signals in the VR app developed in Unity. Respiratory sounds (red) and flow (green). The horizontal and vertical bars allow to modify the time expansion and amplitude gain, respectively

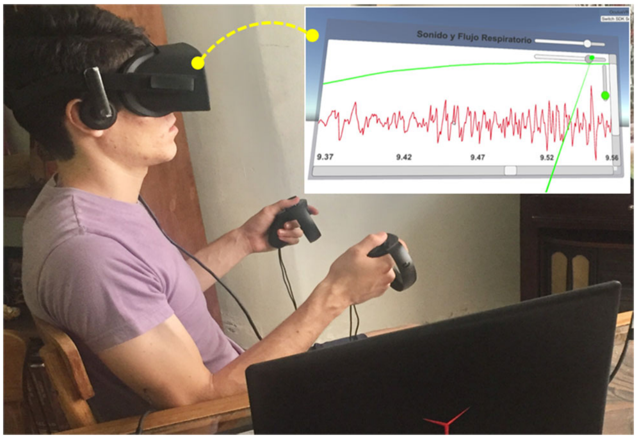

Fig. 4 Illustration of the interactive manipulation of the signals using the VR app and Oculus Rift system

Currently, we are attempting to implement additional basic signal processing functions in the VR app, including spectral analysis and time-frequency analysis, among others. As a future work, we intend to improve the system's interactivity capabilities to allow users to manipulate RS signals by means of more natural movements (i.e. gestures) while the controllers are handled. These controllers are intuitive and convenient when interacting with RS signals. Finally, we will also seek to incorporate audio features from the Oculus headset to allow user physicians to perform an audiovisual analysis of respiratory sounds.

5 Conclusion

VR applications are popularly exploited for industrial, educational, and entertainment purposes, among others. In medical field, VR applications are popular for teaching and training. Efforts such as this research work have the potential to expand VR applications in medical field, particularly in terms of biomedical signal analysis, for both research purposes and clinical practice. Our VR app aims primarily at RS signals, yet it could be used to manipulate any other type of biomedical signal.

Compliance with ethical standards All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Pasterkamp H, Carson C, Daien D, Oh Y. Digital respirosonography. New images of lungsounds. Chest. 1989;96:1405–12.
2. Murphy RL, Holford SK, Knowler WC. Visual lung-sound characterization by time-expanded waveform analysis. N Engl J Med. 1977;296:968–71.
3. Kiss G. Using Smartphones in Healthcare and to Save Lives. In: 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing. IEEE, Dalian, China, pp 614–619. 2011.
4. Quintero-Guerrero CD, Sierra-Ballen EL, Sarmiento-Manrique WJ. Diseño de un Prototipo de Sistema de Realidad Virtual Inmerso Simplificado. Ciencia e Ingeniería Neogranadina. 2008;18:35–50.
5. Bowman DA, Hodges LF, Allison D, Wineman J. The educational value of an information-rich virtual environment. Presence Teleop Virt. 1999;8:317–31.
6. Huang YJ, Fujiwara T, Lin YX, Lin WC, Ma KL. A gesture system for graph visualization in virtual reality environments. In: 2017 IEEE Pacific Visualization Symposium (PacificVis). IEEE, Seoul, South Korea, pp 41–45. 2017.
7. Ouazzani I. Manual de Creación de Videojuego en Unity 3D (Master's tesis). Universidad Carlos III de Madrid. 2012.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.