

PAPER

Improving Monitoring of Heart Rate Using an RGB Camera and OpenCL Architecture: Towards a Heterogeneous Embedded System Implementation

Zakaria El Khadiri¹(✉),
Rachid Latif¹, Amine Saddik^{1,2}

¹Laboratory of Systems Engineering and Information Technology (LISTI), National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco

²Faculty of Applied Sciences, Ibn Zohr University, Ait Melloul, Morocco

zakaria.elkhadiri@edu.uiz.ac.ma

ABSTRACT

Conventional heart rate (HR) monitoring typically relies on contact sensors, but recent advancements demonstrate the potential of non-contact methods using RGB cameras for photoplethysmography (PPG)-based HR analysis. This study presents a real-time, non-contact HR monitoring system that applies signal processing techniques to accurately derive HR from facial video data. Our approach mitigates environmental and motion-induced noise through image enhancement and signal filtering while utilizing Fourier analysis to extract physiological signals from the processed PPG data. Implemented on a heterogeneous CPU-GPU system with high-level synthesis (HLS) for parallel acceleration, our proposed system achieves a substantial improvement in processing efficiency, outperforming the baseline method by a factor of 3.53 in processing time. These results underscore the system's potential for integration into embedded healthcare monitoring applications, offering a pathway for reliable, non-invasive physiological monitoring.

KEYWORDS

heart activity, embedded systems, heterogeneous architecture, CPU-GPU, OpenCL

1 INTRODUCTION

Heart rate (HR) is the number of heartbeats per unit (usually one minute). It is a numerical concept that may also be described as the number of cycles per second [1]. At rest, the HR of healthy adults typically ranges from 60 to 90 beats per minute [2–4]. The resting HR is not constant throughout the day due to several biological cycles; it peaks at noon and rises in response to digestion, heat, and cold. Additionally, a variety of factors, including stress, emotions, anxiety, body temperature, and lack of sleep, among others, can induce an increase or decrease in HR. The traditional methods for measuring HR rely on contact sensors, which are valid in situations

El Khadiri, Z., Latif, R., Saddik, A. (2025). Improving Monitoring of Heart Rate Using an RGB Camera and OpenCL Architecture: Towards a Heterogeneous Embedded System Implementation. *International Journal of Online and Biomedical Engineering (iJOE)*, 21(2), pp. 63–83. <https://doi.org/10.3991/ijoe.v21i02.52203>

Article submitted 2024-09-13. Revision uploaded 2024-11-16. Final acceptance 2024-11-18.

© 2025 by the authors of this article. Published under CC-BY.

where there is no risk to the doctor's and patient's touch, which is not the case today for those who have been exposed to COVID-19 or any other contagious disease. Therefore, several studies were conducted to categorize the cardiac data of patients using image and signal processing techniques. In this context, Hassan et al. [5] present a study on the viability of a non-contact monitoring system for extracting HR. This study established the use of RGB cameras to record momentary changes in heart activity while considering the presence of illumination, motion, skin tone, and distance variance. On the other hand, Zhao et al. [6] proposed a novel system based on preliminary pulse separation and utilizing cascaded least mean square adaptive filters to eliminate modeling errors created during HR extraction. The system was deployed on a treadmill to track athletes. Based on an actual prototype used to evaluate the results, the authors were able to increase the accuracy of HR measurements by an average of more than 10% when compared to current algorithms.

The current paper describes an enhanced pulse activity extraction system that is implemented in a heterogeneous system (CPU-GPU) using the high-level specification programming language, which enables the use of advanced language features to create an optimal and effective microarchitecture [7]. In this instance, we will rely on parallel programming with OpenCL. The paper of Rouast et al. [8, 9], which presents an effective HR monitoring algorithm based on non-contact approaches, served as the foundation for our system. Additionally, a detailed analysis of the hardware-software co-design application will be required to suggest an effective extraction pattern. The complexity, computing process, and execution time of the algorithm will be the main criteria for dividing the algorithm into three main blocks, each of which has around two to three functional blocks (FB). Overall, our study introduces several novel contributions and significant modifications. Firstly, we conducted a detailed analysis of the algorithm, splitting it into function blocks to facilitate targeted optimization efforts. This approach led to notable improvements in processing time within specific FBs. Additionally, we provided a comprehensive comparative analysis between homogeneous and heterogeneous architectures, shedding light on the processing time achieved by the homogeneous architecture using the CPU and by the heterogeneous system based on CPU-GPU, respectively. Furthermore, we proposed an efficient pattern model dedicated to heterogeneous system implementation to optimize system resources, processing time, and maximize performance. The following is a summary of our contribution:

1. Processing time improvement in some functional blocks.
2. Reviewing the HR real-time monitoring algorithm using remote photoplethysmography (rPPG) analysis.
3. Comparative analysis between the homogeneous and heterogeneous architecture.
4. Providing an efficient pattern model dedicated to heterogeneous system implementation.

The remainder of this document is organized as follows: In Section 2, we introduce the state-of-the-art methods for physiological parameter estimation using a webcam and describe the proposed architecture. Section 3 presents the results obtained from the heterogeneous architecture. Finally, Section 4 discusses future work. The conclusion is formulated in Section 5.

2 METHODS

2.1 State-of-the-art

The cardiovascular system is essential for blood circulation throughout the human body, with color variations on facial skin during each heartbeat reflecting this activity. Extracting HR from these color fluctuations has emerged as a promising non-contact method, enhancing our understanding of cardiovascular function. Since Costa et al. [10] introduced the first remote heart monitoring system, research has focused on using camera images to obtain physiological data. Over recent years, algorithms estimating blood volume pulse (BVP), HR, and respiratory rate (RR) from human faces through photoplethysmography (PPG) analysis have significantly evolved. Scebba et al. [11, 12] advanced this field by employing multispectral data fusion to estimate respiratory rate (RR), particularly in apnea cases. Their approach utilized synchronized far-infrared (FIR) and near-infrared (NIR) cameras, chosen for their ability to operate without visible light. Despite challenges such as low pixel resolution and sampling rate, their algorithm performed a three-step process: locating feedback signals from nostrils and thorax, extracting respiratory signals from the region of interest (ROI), and applying a fusion model combining RR estimation and an apnea clustering approach. Evaluated on data from 30 healthy adults, the algorithm performed well in estimating RR during controlled respiration tasks, even in the presence of central and obstructive apneas.

Several studies have since proposed neural network-based methods for estimating respiratory frequency, often utilizing PPG and ECG signals or thermal sequences to enhance non-contact RR accuracy. Some methods focus on extracting RR from respiratory quality indices combined with neural networks [13], while others process thermal sequences with deep neural networks for improved RR estimation [14, 15]. Most studies utilize a single camera to monitor BVP from skin regions such as the face, hand, or foot. However, Ghanadian et al. [16] demonstrated that multiple cameras could increase HR estimation accuracy, especially in mobile subjects. Their algorithm applied blind source separation (BSS) and independent component analysis (ICA) to separate HR signals, proving that multiple camera feeds significantly enhance HR accuracy compared to single-camera approaches. Further, Moya-Albor et al. [17] proposed a novel non-contact respiratory rate estimation technique using Eulerian motion magnification and a convolutional neural network (CNN). They compared two CNN-based methods: a CNN-ROI approach requiring manual ROI selection and a CNN whole-image method that requires no ROI. The system achieved a mean absolute error of $3.28 \pm 3.33\%$ for automatic RR estimation, showing successful results across various body positions and environments. El Khadiri et al. [18, 19] introduced a comprehensive signal processing framework for extracting HR and RR, employing advanced decomposition techniques such as empirical mode decomposition (EMD) and empirical wavelet transform (EWT). Their approach includes principal component analysis (PCA) for dimensionality reduction and the Moving Average method for noise reduction, providing a robust solution for non-contact monitoring in various conditions [20–28].

Our study builds on these advancements, further addressing limitations in existing approaches. Traditional non-contact HR monitoring often faces challenges with motion artifacts, variable lighting, and inconsistent signal quality, affecting real-world reliability. To overcome these issues, we integrate several preprocessing steps, including grayscale histogram equalization and ROI extraction, to reduce noise and clarify the signal. We then apply moving average filtering and PCA to further isolate pulse activity data from facial images, providing a clearer signal for HR analysis. A key innovation in our approach is the use of a heterogeneous CPU-GPU system for real-time processing.

By employing high-level synthesis (HLS) as a parallel programming tool, we efficiently distribute computational tasks across CPU and GPU resources, optimizing processing time and achieving a 3.53-fold improvement in speed compared to existing naive implementations. This heterogeneous setup enhances real-time feasibility, making our method more suitable for continuous HR monitoring in practical applications.

Furthermore, our study on non-contact physiological monitoring shares foundational principles with diverse research efforts focused on real-time signal processing and adaptive control across applications, including autonomous vehicles, robotics, and medical control systems. For instance, autonomous vehicle research emphasizes edge computing and vehicle-to-everything (V2X) communication to handle heterogeneous sensor data, ensure redundancy, and meet strict safety standards [29–34]. Similarly, our system enhances computational efficiency and accuracy through a heterogeneous CPU-GPU framework optimized for real-time physiological monitoring, echoing similar computational strategies in real-time vehicle and robotics applications [35]. Additionally, medical control methodologies, such as adaptive neural control for cancer treatment [36–39] and discrete-time state feedback control for regulating HR during exercise [40, 41], demonstrate adaptive responses to biological data. Our study aligns with these by employing preprocessing, ROI extraction, and noise-reduction strategies to manage challenges unique to non-contact physiological monitoring, such as motion artifacts and lighting variability. This comparison highlights how advancements in one domain, such as cross-disciplinary methodologies for real-time adaptive systems, can inspire innovations in non-invasive health monitoring.

2.2 Proposed architecture

As aforementioned, in the current paper, we will present a new optimized pattern for HR estimation based on non-contact algorithms, PPG signal analysis, and heterogeneous system computing. Our study builds upon the work of Rouast et al. [8], who proposed a HR monitoring algorithm using non-contact techniques and an RGB camera for PPG signal analysis. These techniques will aid in gathering the necessary pulse activity data from facial image sequences on the first hand, independent of the illumination of the environment and mitigating undesired noises from the PPG signal. In line with the hardware-software co-design concept, our proposed architecture will be integrated into a heterogeneous system. The utilization of heterogeneous computing is prevalent in engineering, the big data field, and science to address computing performance requirements and real-time constraints.

Discussion of heterogeneous systems often involves considering various acceleration devices, including GPUs, FPGAs, DSPs, and others. However, the latter devices play a significant role as coprocessors for intensive parallel arithmetic workloads due to their heterogeneous architecture (CPU-GPU, CPU-FPGA, CPU-DSP, etc.). Several programming tools are available for parallel computing. The most common tool is OpenCL (Open Computing Language), proposed by the Khronos Group, which combines an API and a programming language derived from C. It is used to program heterogeneous parallel systems that include, for example, both a multi-core CPU and a GPU [42]. Additionally, there is another programming tool called CUDA, which utilizes a GPGPU (General-Purpose Graphics Processing Unit), enabling the GPU to perform general-purpose calculations instead of the CPU. These processors typically consist of around a thousand computation circuits, typically operating at 1 GHz, representing a much higher potential than a central processor at 4 GHz. In our case, we will focus particularly on CPUs and GPUs using OpenCL as the language for parallel computing. Furthermore, Table 1 provides a brief comparison between CUDA and OpenCL concepts, and Figure 1 depicts

the heterogeneous nature of OpenCL, which comprises two distinct components: the Host side, representing the CPU with multiple arithmetic execution units, and the Device side, encompassing the GPU, FPGA, or other acceleration devices.

Table 1. The comparison between CUDA and OpenCL concepts

Type	OpenCL	CUDA
Developer	By Khronos	By Nvidia
Tooling support	Open Source	Proprietary Solution
Compute device	Compute Units	Streaming multiprocessors
Index space	NDRange	Grid Index
Execution model	Work-Group/Work items	Thread-block/Thread
Fundamental memory	Global/Local/Private memory	Global/Shared/Local memory
Scalar processor	Processing element	Scalar core

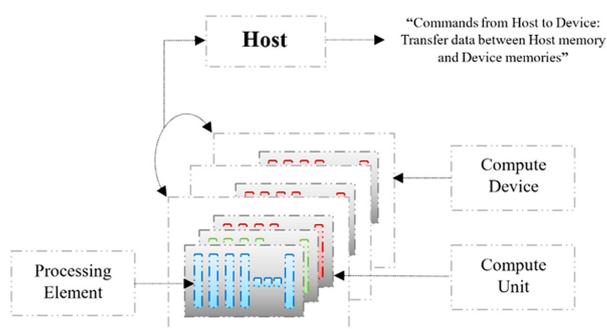


Fig. 1. OpenCL architecture

The current algorithm is based on a collection of techniques used specifically in the signal and image processing fields. These techniques often perform functions such as image preprocessing, signal filtration and decomposition, feature and characteristic extraction from the input image or signal, and domain conversion. Figure 2 provides a summary of the various techniques used to estimate HR for your convenience.

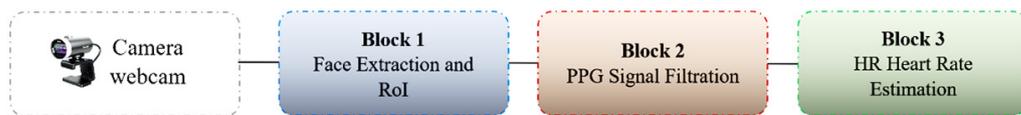


Fig. 2. General algorithm for heart rate estimation

Figure 2 presents an overview of the general algorithm utilized for HR estimation in our study. The algorithm incorporates various techniques and processing steps aimed at accurately determining the HR from physiological signals captured via camera imaging. The algorithm begins with the acquisition of physiological signals, typically obtained from facial regions using RGB camera technology. Subsequently, preprocessing steps, such as noise reduction and signal enhancement, are applied to enhance the quality of the acquired signals. Following preprocessing, the algorithm employs feature extraction methods to identify relevant signal characteristics indicative of HR variations. Once features are extracted, the algorithm utilizes signal processing techniques, such as Fourier analysis, to further analyze the signal and extract HR information. This may involve frequency domain transformations to reveal the periodicity of the signal corresponding to the HR. The method used in this work is generally divided

into three main blocks: the first main block (B1) is for the extraction of human faces along with their ROI; the second main block (B2) is for plethysmography signal filtration and extraction of relevant components; and the third block is for the estimation of HR value. This subdividing has been evenly distributed based on the complexity of each algorithm. The three primary blocks for the entire system are shown in Figure 3.

Figure 3 illustrates the comprehensive architecture comprising various function blocks inherent in each primary block, along with a depiction of the inputs and outputs associated with each function block. Figure 3 provides an overview of the hierarchical structure of our proposed system, highlighting the interconnections and dependencies between different components. Each main block encapsulates distinct function blocks, responsible for specific tasks within the system. As depicted in Figures 2 and 3, three main blocks (B1, B2, and B3) can be identified. Each of these blocks is composed of two to three FBs. The first block (B1) is centered on face and ROI extraction. This block consists of three FBs: the first (FB1) facilitates the extraction of face image sequences, the second (FB2) undertakes image pre-processing (grayscale conversion, image equalization), and then the third (FB3) is utilized for extracting the Regions of Interest (RoI). The second block (B2) is regarded as the most critical because it encompasses PPG signal normalization, filtration, and detrending. This block also comprises three FBs: FB1 for extracting the PPG signal from the filtered RoI sequences, FB2 for signal filtering (normalization, moving average filter, detrending), and the final FB (FB3) for applying the Principal Component Analysis (PCA) method to the PPG signal. Finally, the third block (B3) is for HR estimation. This block is subdivided into two FBs: the first one (FB1) has the role of transforming the signal into the frequency domain using the discrete Fourier transform (DFT), while the second one performs the calculation of the HR value.

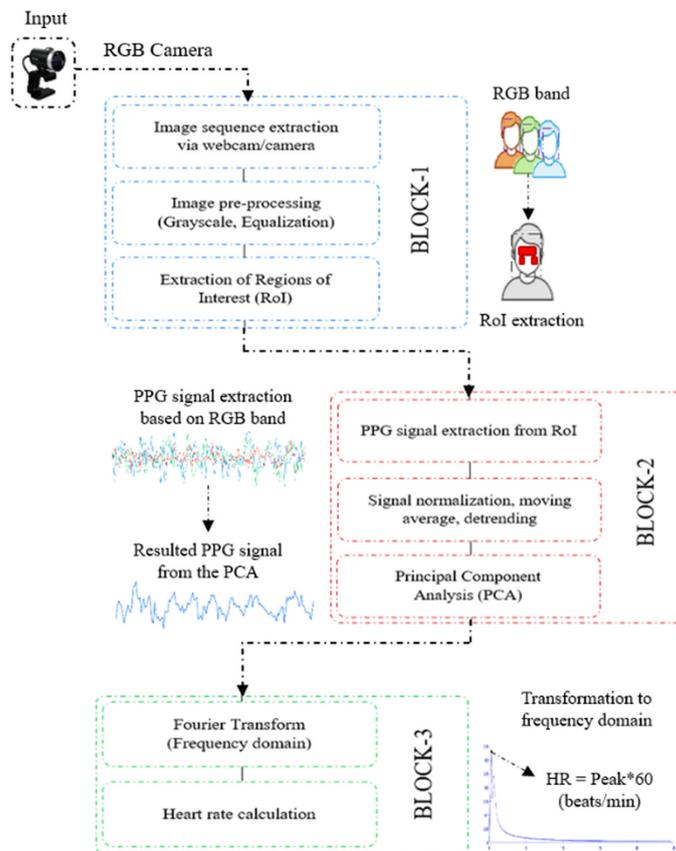


Fig. 3. The overall architecture of the different function blocks in each main block

Specifically, the B1 block is dedicated to housing a multitude of image processing algorithms aimed at extracting various sequences from the user's images and facilitating the extraction process of ROI. This involves utilizing the well-known Viola and Jones method [43], which relies on cascades of boosted classifiers. On the other hand, B2 occupies the major part of this algorithm; indeed, this block contains more processing compared to the other blocks, as it produces the RGB signals by averaging the values of the R, G, and B band frames. Subsequently, the obtained signal must be normalized, according to the method mentioned [44]. This normalization is carried out using the following equation (1):

$$Y_i(t) = \frac{X_i(t) - \mu_i(t)}{\delta_i} \tag{1}$$

For each i (= R, G, and B) signal, X_i is the input brute PPG signal, Y_i is the normalized PPG signal, μ_i is the mean, and δ_i is the standard deviation of Y_i .

The second step of this block involves eliminating various undesirable trends in the series, notably environmental parameters such as temperature or external noise. This step incorporates a moving average filter (equivalent to a low-pass filter) and the detrending method [45] (equivalent to a high-pass filter), which is based on the smoothness prior's approach. The FB3 focuses on the Principal Component Analysis (PCA) method [46]. PCA is a method within the data analysis family that involves transforming interrelated (correlated) variables into new variables that are decorrelated from each other; these new variables are termed principal components. This method separates the raw signals into linearly uncorrelated components and orders them based on variance. In practice, the PCA algorithm relies on significant mathematical concepts, including variance and covariance, eigenvalues, and eigenvectors. Moreover, considering X as the input signal (with n as the signal's dimension), the fundamentals of the PCA method are developed in several steps (see Algorithm 1) as follows, and it can be resumed with the following pseudocode:

$$X(t) = [X_1(t), \dots, X_n(t)] \tag{2}$$

Compute the standardization input dataset X' , the covariance matrix Cov , and the eigenvector U :

$$X'(t) = \frac{[X - \mu]}{\delta} \tag{3}$$

$$Cov = X'^T X' = V \Lambda V^T \tag{4}$$

$$U = X' V \Lambda^{-1/2} \tag{5}$$

Where: X' the standardization dataset, μ is the mean value of the input signal X , δ is the standard deviation of the input signal X .

Ordering the eigenvectors accordingly of eigenvalues and calculating the principal components:

$$Y = [U_1, \dots, U_i] \tag{6}$$

$$PCs = Y^T X' \tag{7}$$

Where the PCs are the principal components in order of feature importance.

Algorithm 1: PCA

```

Start Algorithm
1. // Principal Component Analysis – PCA
2. // The PCA function has the following parameters:
3. // Input  $X(t)$ : Set of PPG signal (n-dimension):
4. //  $X(t) = [X_1(t), \dots, X_n(t)]$ .
5. // Output: New Principal Components:  $P(t)$ .
6.  $P_i(t) = \text{PCAFunction}(X(t))$ 
7. // Dataset standardization  $X'(t)$ 
8.  $X'(t) = [X - \text{mean}(X)] / \text{standard deviation}(X)$ ;
9. // Covariance matrix computing
10.  $\text{Covariance} = X'^T X' = \bar{V} \Lambda \bar{V}^T$ ;
11. // Eigenvector computing  $U$ 
12.  $U = X' V \Lambda^{-1/2}$ ;
13. // Ordering the Eigenvectors accordingly of Eigenvalues
14.  $Y = [U_1, \dots, U_i]$ ;
15. // Principal Components computing  $P$ 
16.  $P = Y^T X'$ ;
17. END Function
END Algorithm
    
```

Finally, the B3 is dedicated to HR estimation. Initially, a signal conversion from the time domain to the frequency domain is performed using the Fourier transform method. Subsequently, the HR value can be estimated by correlating it with the index equivalent to the highest spectral power. After dividing the entire algorithm into several function blocks, each one will be evaluated independently to integrate into a heterogeneous system. This integration will be based on factors such as processing time and the process complexity of each FB. The overall suggested pattern for execution by the host (CPU) and the device (GPU) can be illustrated as follows:

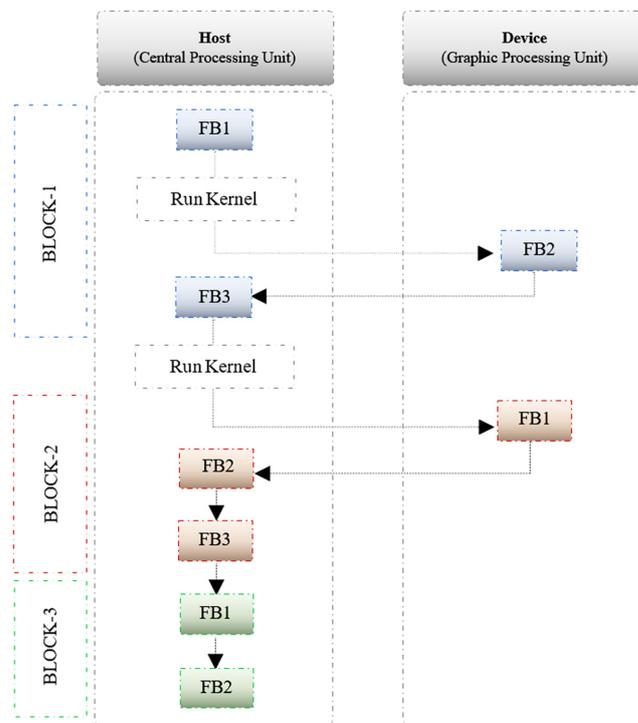


Fig. 4. The overall architecture executed in host and device parts

Figure 4 illustrates the comprehensive architecture of function blocks within each main block, highlighting the input and output connections of individual function blocks. The depicted architecture showcases the integration of various computational elements and processing stages, each serving a specific function in the overall system. In particular, the figure delineates the execution and implementation of these function blocks across both CPU and GPU platforms, emphasizing the parallel processing capabilities of the GPU architecture. By utilizing both CPU and GPU resources, the system achieves optimized performance and efficiency in processing the input data. As outlined in Figure 4 above, the first component to be executed on the GPU side is the B1-FB2 FB. Indeed, several phases need to be considered before executing the entire computation in the device component. Firstly, an OpenCL variable declaration function served to determine the platform ID and device ID to define the kernel's argument requirement. Then, the necessary buffers are created, which will serve as the interconnection between the data gathered from the CPU part and the GPU side (such as the pixels of the image). After sending the request to execute the kernel and retrieving the result from the GPU side once the computation is finished, the next step involves extracting the ROI from the obtained result. This process transfers the relevant data from the GPU to the CPU portion. Afterward, the identical stacking steps will then be executed by FB1 in the second main block to compute the mean value and estimate the noisy PPG signal version. The stacking steps necessary to perform the B1-FB2 and B2-FB1a are illustrated by the following pseudo-code (see Algorithm 2):

Algorithm 2: B1-FB2 and B2-FB1

Start Algorithm

1. // B1-FB2 → Image pre-processing through grayscale conversion technique
2. // B2-FB1 → Mean calculation for PPG signal extraction
3. Function OpenCL_variable_declaration()
4. // Variable declaration needed for the interconnection between CPU and GPU
5. platform_id, device_id, context, command_queue, program, kernel;
6. // Declaration of the memory object and getting the platform and the device IDs
7. Rmobj, Gmobj, Bmobj, input buffer, reductionBuffer;
8. clGetPlatformIDs(&platform_id);
9. clGetDeviceIDs(&device_id);
10. // Creating the context, the command queue, and the buffers
11. clCreateContext(&device_id);
12. clCreateCommandQueue(context);
13. [Rmobj, Gmobj, Bmobj] ← clCreateBuffer();
14. [inputBuffer, reductionBuffer] ← clCreateBuffer();
15. clEnqueueWriteBuffer();
16. // Creating the kernel and setting the kernel arguments
17. kernel ← clCreateKernel();
18. clSetKernelArg();
19. END function
20. // Run the two kernels for B1-FB2 and B2-FB1 parts
21. clEnqueueNDRangeKernel();
22. // Retrieve the result from the GPU side
23. clEnqueueReadBuffer();

END Algorithm

The following Figure 5 depicts the header codes executed in the device part for B1-FB2 and B2-FB. The figure illustrates the pseudo-codes representing the headers' function executed within the Device component for two distinct processes: B1-FB2 and B2-FB1. The headers' function plays a crucial role in managing and processing

data within the embedded system architecture proposed in this study. In the B1-FB2 process, the pseudocode demonstrates the sequence of operations involved in processing incoming data. Similarly, for the B2-FB1 process, the pseudo-code outlines the steps for processing received data headers and extracting relevant information for further processing.

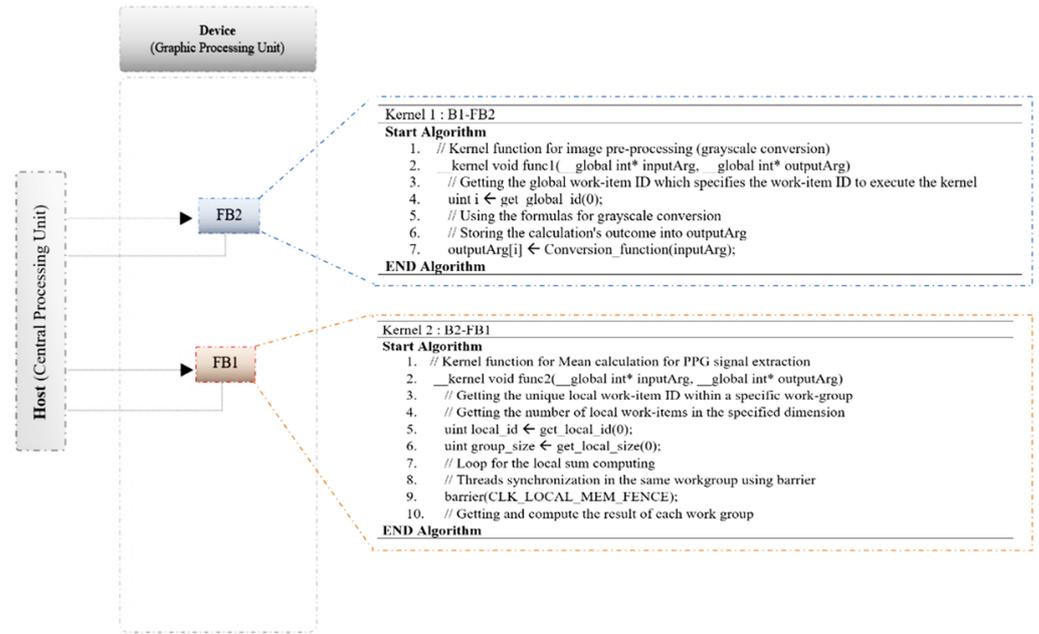


Fig. 5. Pseudo-codes for the headers function executed in the device part

3 RESULTS

After defining the different FBs, it is necessary to study the possibility of implementing some of them in a heterogeneous architecture, such as a CPU-GPU system, using OpenCL parallel programming. Therefore, to separate these FBs between the host and the device parts, we will analyze each one, including its execution and treatment time. We will handle the blocks that consume a lot of time in the device part. Utilizing a system based on a GPU platform will accelerate the algorithm's processing and reduce the treatment time, thereby aiding in monitoring the physiological human state with a real-time constraint. Time evaluation has shown that the second FB of B1 and the first FB of B2 consume more time than other blocks, which is logical due to the calculation amount and the pre-processing performed on the extracted images, including grayscale conversion, image equalization, PPG signal normalization, and moving average filtration. Therefore, we can compute both of them with GPU acceleration (see Figure 3). This acceleration will reduce the global execution time for each block. Figure 6 illustrates the stacking and global architecture of these blocks.

Figure 6 presents a comprehensive overview of the heterogeneous system architecture, showcasing the arrangement and interaction of function blocks within the CPU-GPU configuration. This diagram provides a visual representation of how tasks and computations are distributed and processed across the CPU and GPU components. Understanding the stacking of function blocks is crucial for comprehending the overall functionality and performance optimization strategies employed in the

proposed system. In our experimental research, we utilized several PPG signals extracted from participants to evaluate variations in HR values, considering significant noise sources such as user head movement, ambient lighting fluctuations, and environmental interferences. The normalized and denoised PPG signals derived from our experimental inquiry are depicted in Figures 7–10.

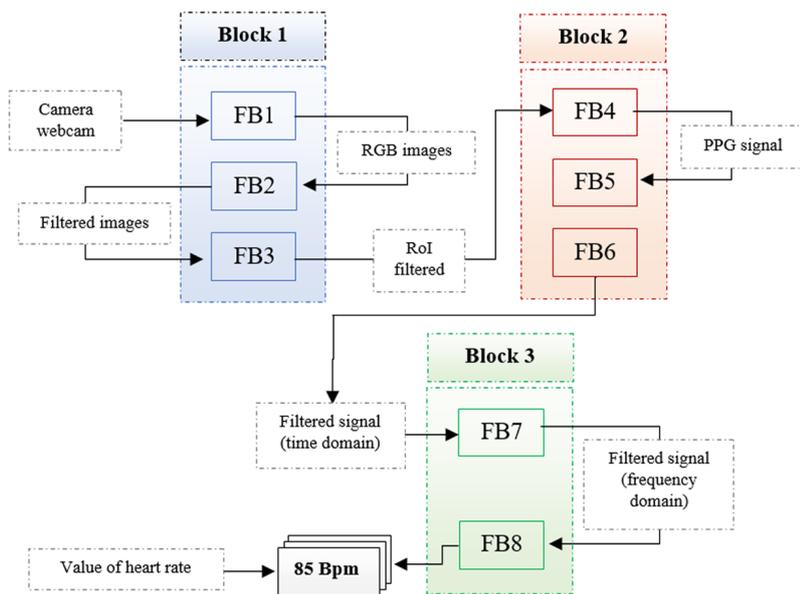


Fig. 6. Global diagram of the stacking of function blocks

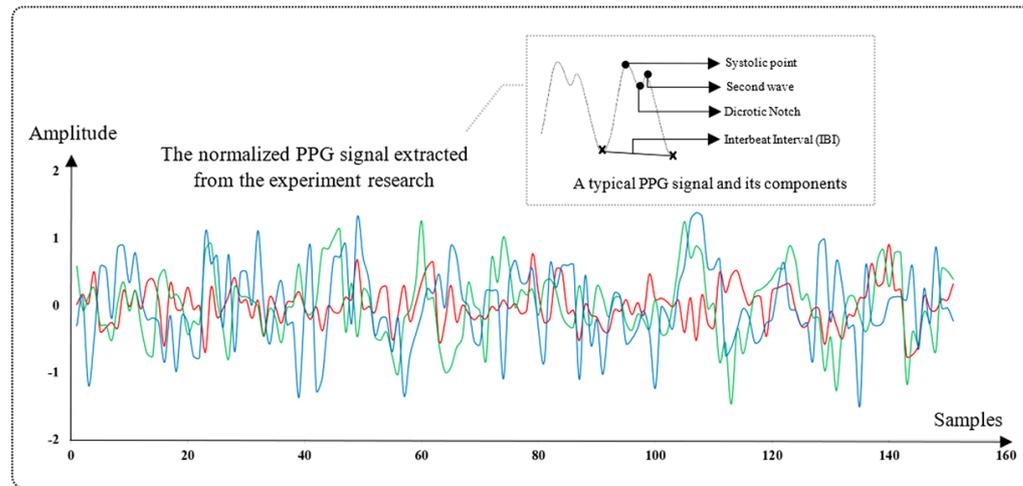


Fig. 7. The normalized PPG signal extracted from the experimental research

Figure 7 illustrates the normalized photoplethysmogram (PPG) signal obtained from the experimental research conducted in our study. The normalized PPG signal is compared to a typical PPG signal and its components, providing insights into the variations and characteristics observed in our experimental data. The components of the typical PPG signal may include the systolic peak, diastolic peak, and dicrotic notch, among others. By comparing the normalized PPG signal with the typical PPG signal and its components, we aim to highlight any deviations or unique features present in our experimental findings. This comparison aids in understanding the effectiveness and reliability of our proposed methodology for PPG signal extraction and analysis.

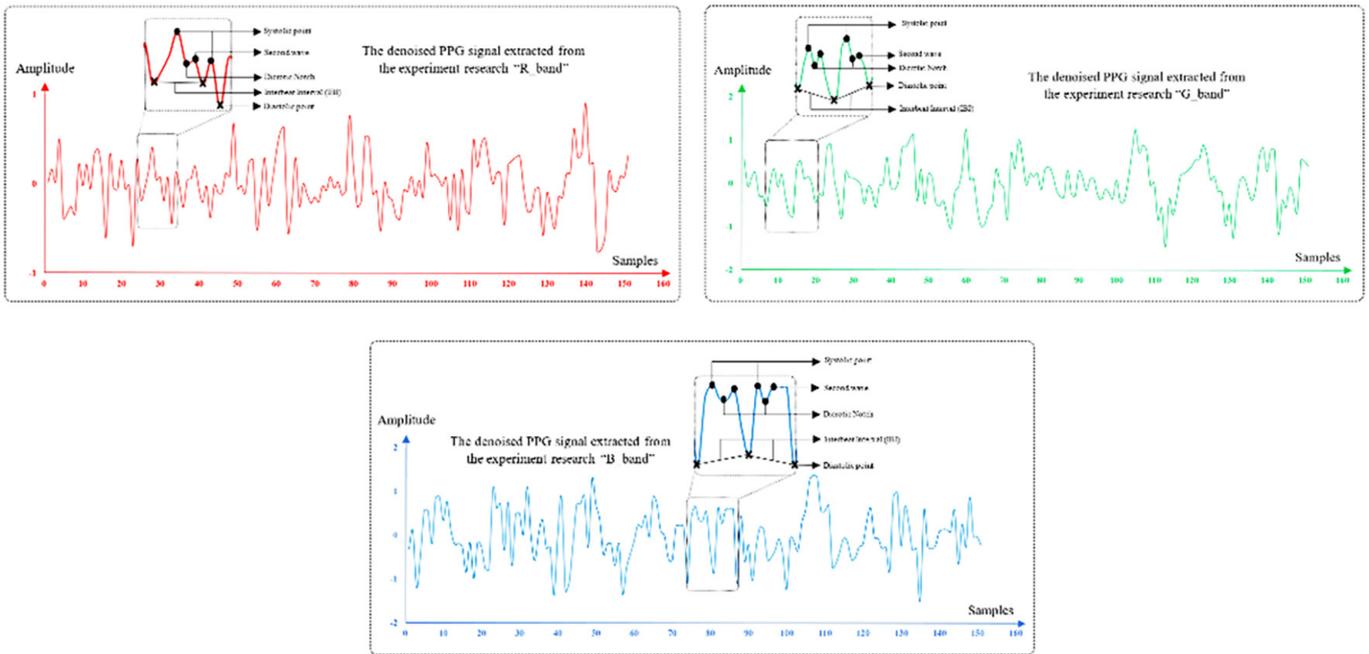


Fig. 8. The denoised PPG signal extracted from the experimental research “RGB_band”

Figures 8 depict the denoised PPG signal extracted from the experimental research within the red, green, and blue color space representations, respectively. The denoising process applied to the PPG signal aims to reduce noise and artifacts, thereby enhancing the quality and reliability of the signal for subsequent analysis. The utilization of the RGB color space representation provides valuable insights into the characteristics of the PPG signal, allowing for a detailed examination of physiological variations. Despite the substantial variables of the noises intentionally generated, the method used to extract the HR values from multiple PPG signals yields numerous findings and results that are relatively similar. These findings demonstrate that the signal processing and image preprocessing methods employed in the current system can effectively handle unexpected noise and mitigate undesired effects. Our analysis reveals mean HR values of approximately 76.3 BPM, 74.38 BPM, 80.56 BPM, and 72.18 BPM, respectively, for the four PPG signals, which converge roughly towards the same value. Figures 9 and 10 depict distribution histograms for the four PPG signals.

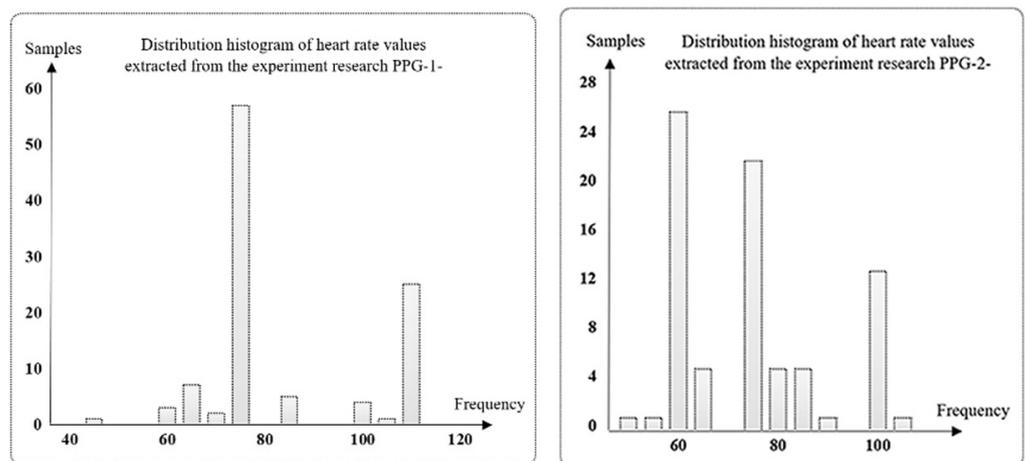


Fig. 9. The distribution histogram of the heart rate values extracted from PPG 1 and 2

Figure 9 illustrates the distribution histogram of HR values extracted from experimental research conducted using PPG 1. The histogram reveals significant peaks within the range of 70 to 80 beats per minute (bpm) and another prominent peak around 110 bpm. These findings provide valuable insights into the distribution pattern of HR values obtained from PPG 1 data. In the same figure, we present the distribution histogram of HR values extracted from experimental research conducted using PPG 2. Significant peaks are observed within the range of 60 to 90 bpm, with another notable peak around 100 bpm. The histogram offers insights into the distribution pattern of HR values obtained from PPG 2 data, providing valuable information for the analysis of cardiovascular dynamics in the experimental setting.

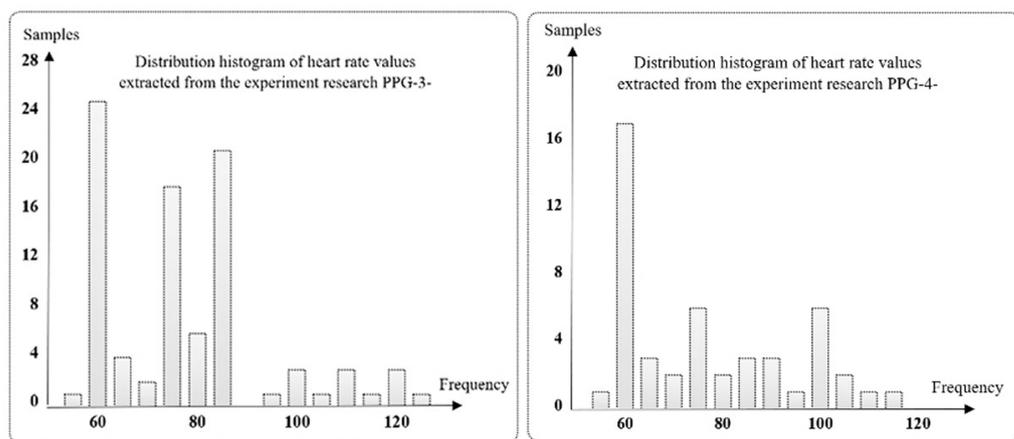


Fig. 10. The distribution histogram of the heart rate values extracted from PPG 3 and 4

The above histogram illustrates the distribution of HR values extracted from experimental research conducted using PPG 3. The histogram reveals significant occurrences of HRs falling within the range of 60 to 90 bpm, indicating a prevalent frequency of HRs within this range. Conversely, lower occurrences are observed within the range of 100 to 120 bpm. Similarly, Figure 10 showcases the distribution of HR values extracted from experimental research conducted using PPG 4. Notably, the histogram displays a significant concentration of HR values around the 60-bpm mark. The distribution pattern depicted in this histogram offers valuable insights into the prevalence of specific HR values derived from the experimental data collected using PPG 4.

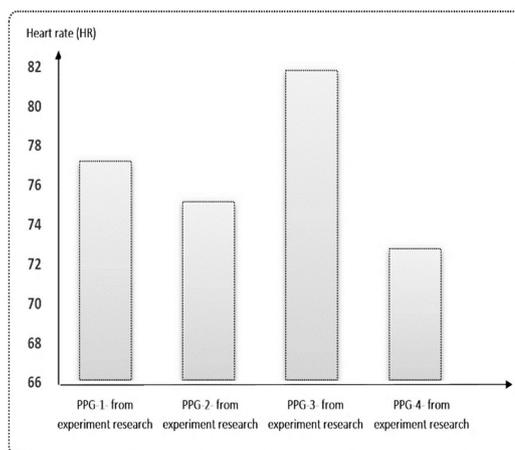


Fig. 11. Grouped histogram of the estimated heart rate from the experimental research

Figure 11 depicts a grouped histogram presenting data pertinent to our study. Specifically, the histogram visually represents the distribution of HR values corresponding to each utilized PPG signal. This histogram was derived by aggregating data collected from our experimental setup, which employed RGB cameras for capturing PPG signals. The experimental data utilized in our study were obtained from a diverse database comprising more than 20 PPG signals extracted from adult individuals participating in the experiment. The sample consisted of volunteers of varying ages, genders, and skin colors, ensuring a representative range of physiological characteristics. Furthermore, the experiment durations varied from 20 seconds to over 10 minutes, capturing a wide spectrum of physiological responses under different conditions. Additionally, it's worth noting that the experimental environment was deliberately perturbed by fluctuating illumination and inconsistent lighting conditions, simulating real-world scenarios. The range of light intensity used during the experiments was between 100 and 200 lux, reflecting typical indoor lighting conditions. In our experimental research, the system used in this implementation is based on an Intel Core i5-4200U CPU (2.30 GHz) with 4 GB of RAM and a graphics card: Advanced Micro Device (AMD) with a 975 MHz core (max. compute units: 5, max. work item dimensions: 3, max. work item sizes: 1024, max. work group sizes: 256). As outlined above, the temporal study of the algorithm has shown that blocks two and four are recommended to be sent to the GPU part to accelerate the processing; this choice depends on the consumption time of each FB. In this case, it is necessary to calculate the processing time of each block to understand the algorithmic constraints summarized in the dependency of each input variable for each block. For convenience, Table 2 illustrates the execution time of each function block in each main block.

Table 2. The execution time of each functional block in the host and device parts

Functional Blocks (FBs)		Execution Time in CPU	Execution Time in CPU-GPU
B1	FB2	≈ 0.500 ms	≈ 0.004 ms
	FB4	≈ 2.880 ms	≈ 0.415 ms
B2	FB1	≈ 0.414 ms	–
	FB2	≈ 0.175 ms	–
	FB3	≈ 0.050 ms	–
B3	FB1	≈ 0.109 ms	–
	FB2	–	–

As shown in the table above, two FBs, specifically the second and fourth FBs (FB2 and FB4), consume more time compared to the remaining blocks. In a homogeneous system (CPU), FB2 takes 0.5 ms, and FB4 takes 2.88 ms, while the other neighboring FBs consume less than 0.45 ms, making them ineligible for the current optimization. Figure 12 illustrates that the first block executed by the GPU is FB2 of the first main block, which takes approximately 0.004 ms in a heterogeneous system (CPU-GPU), 1.25 times faster than homogeneous computing. Conversely, FB1 of the second main block takes 2.88 ms in a homogeneous system, while its computing times take about 0.415 ms in a heterogeneous system, resulting in an acceleration rate of 6.94 times faster than CPU execution. Figure 12 below summarizes the average times for both architectures.

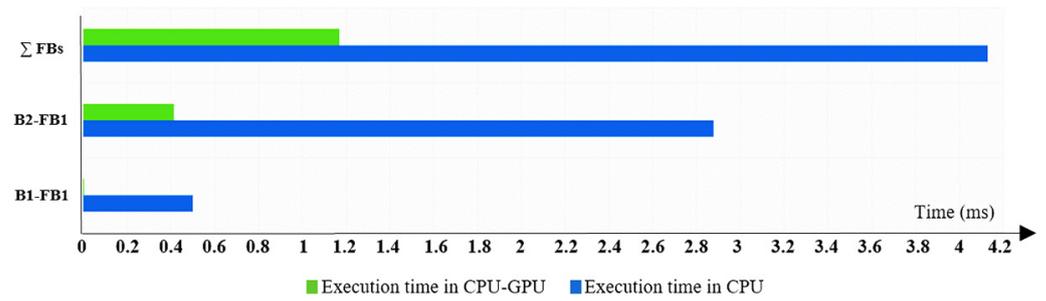


Fig. 12. Histogram of execution times using homogeneous and heterogeneous systems

Figure 12 illustrates a grouped histogram comparing the execution times of a particular task or set of tasks using both homogeneous and heterogeneous computing systems. The comparison between homogeneous and heterogeneous systems provides insights into the performance benefits of leveraging different computational architectures for the given task(s). To evaluate any proposed architecture in a heterogeneous system, it is necessary to calculate the latency rate of the execution of the set of tasks used in this algorithm. Researchers use Amdahl’s Law to calculate this acceleration, which is often utilized in parallel computing to predict theoretical acceleration when using multiple processors; indeed, it’s a formula that provides the theoretical speedup in latency of the processing of tasks at a fixed workload [47, 48]. Amdahl’s argument can be outlined as follows:

$$S_{latency}(s) = \frac{1}{(1 - p) + \frac{p}{s}} \tag{8}$$

Where:

- $S_{latency}$ is the theoretical speedup of the execution of the entire task.
- s is the speedup of the part of the task that benefits from improved system resources.
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

In our case, according to the computing times depicted in Table 2, the heterogeneous architecture based on a CPU-GPU system experienced an acceleration rate of almost 3.53 compared to the architecture implemented in the homogeneous system. The latter is observed in the above-grouped histogram; the heterogeneous system provides optimization in terms of processing times compared to the homogeneous architecture. However, the times gained with the use of CPU-GPU are about 0.1945 ms versus 0.688 ms using a single calculator; this results in an acceleration rate of $\times 3.53$ units for the global algorithm. This acceleration rate gained from the use of CPU-GPUs has a crucial influence and is a major contributor to achieving real-time measurement. Whereas it can serve as a benchmark study for further use of high-level tools and languages such as OpenMP, OpenACC, or CUDA [49–53] to process more frames per second based on the parallelization instructions and desired thread count, considering memory access and latency.

4 FUTURE WORK

While our proposed method achieves promising results in HR estimation using non-contact RGB camera methods, certain hardware-specific challenges remain,

especially in the context of heterogeneous CPU-GPU systems. Although utilizing a heterogeneous CPU-GPU setup enhances processing speed and efficiency, this approach depends on specialized hardware resources that may not be feasible in cost-sensitive or mobile applications. This dependency on high-performance, parallel-processing capabilities can limit the portability of our method to embedded systems with more constrained resources. To address this, future work will focus on optimizing the algorithm to better suit a broader range of embedded architectures. This could involve adapting our design to reduce reliance on GPU-specific acceleration, possibly by implementing lightweight, energy-efficient algorithms compatible with simpler embedded processors. Additionally, exploring alternative parallel programming models or lower-complexity algorithms may offer a more balanced trade-off between speed and resource consumption, making the system more adaptable for real-time applications in mobile health monitoring. This direction not only aims to broaden the applicability of our approach across diverse hardware platforms but also emphasizes improvements in portability, enabling the system to function effectively in varied operational environments.

5 CONCLUSION

This paper presents a method for real-time, non-contact HR extraction using a camera or webcam, utilizing variations in facial skin color caused by heartbeat along with several well-known signal and image processing algorithms. The proposed heterogeneous architecture (CPU-GPU) with OpenCL-based parallel programming achieved a significant improvement in processing time, with a gain of 3.53. This contactless technology holds promise for medical and other applications due to the wide availability of cameras and webcams. However, the system's accuracy may be affected by environmental factors such as varying lighting conditions, motion artifacts, and changes in facial orientation. Future work will address these challenges and expand functionality to include other physiological parameters, such as breathing rate, oxygen saturation, and arterial blood oxygen saturation. We also aim to further enhance real-time performance by implementing the system on a CPU-FPGA embedded architecture using hardware-software co-design, which would allow for a more efficient and prompt implementation.

6 ACKNOWLEDGMENTS

We owe a debt of gratitude to the Ministry of National Education, Vocational Training, Higher Education, and Scientific Research (MENFPESRS) and the National Centre for Scientific and Technical Research of Morocco (CNRST) for their financial support for the project Cov/2020/109.

7 REFERENCES

- [1] R. Avram *et al.*, "Real-world heart rate norms in the health eHeart study," *npj Digit. Med.*, vol. 2, 2019. <https://doi.org/10.1038/s41746-019-0134-9>
- [2] W. B. Kannel, C. Kannel, R. S. Paffenbarger Jr., and L. A. Cupples, "Heart rate and cardiovascular mortality: The Framingham study," *Am. Heart J.*, vol. 113, no. 6, pp. 1489–1494, 1987. [https://doi.org/10.1016/0002-8703\(87\)90666-1](https://doi.org/10.1016/0002-8703(87)90666-1)

- [3] R. F. Gillum, "Epidemiology of resting pulse rate of persons ages 25–74: Data from NHANES 1971–74," *Public Health Rep.*, vol. 107, no. 2, pp. 193–201, 1992.
- [4] J. W. Mason, D. J. Ramseth, D. O. Chanter, T. E. Moon, D. B. Goodman, and B. Mendzelevski, "Electrocardiographic reference ranges derived from 79,743 ambulatory subjects," *J. Electrocardiol.*, vol. 40, no. 3, pp. 228–234.e8, 2007. <https://doi.org/10.1016/j.jelectrocard.2006.09.003>
- [5] M. A. Hassan, A. S. Malik, D. Fofi, B. Karasfi, and F. Meriaudeau, "Towards health monitoring using remote heart rate measurement using digital camera: A feasibility study," *Measurement*, vol. 149, p. 106804, 2020. <https://doi.org/10.1016/j.measurement.2019.07.032>
- [6] C. Zhao, C. L. Lin, W. Chen, M. K. Chen, and J. Wang, "Visual heart rate estimation and negative feedback control for fitness exercise," *Biomedical Signal Processing and Control*, vol. 56, p. 101680, 2020. <https://doi.org/10.1016/j.bspc.2019.101680>
- [7] Y. Sun, K. Amiri, G. Wang, B. Yin, J. R. Cavallaro, and T. Ly, "High-level design tools for complex DSP applications," in *DSP for Embedded and Real-Time Systems*, 2012, pp. 133–155. <https://doi.org/10.1016/B978-0-12-386535-9.00008-1>
- [8] P. V. Rouast *et al.*, "Remote heart rate measurement using low-cost RGB face video: A technical literature review," *Front. Comput. Sci.*, vol. 12, pp. 858–872, 2018. <https://doi.org/10.1007/s11704-016-6243-6>
- [9] P. V. Rouast, M. T. P. Adam, D. J. Cornforth, E. Lux, and C. Weinhardt, "Using contactless heart rate measurements for real-time assessment of affective states," in *Information Systems and Neuroscience*, in Lecture Notes in Information Systems and Organisation, F. Davis, R. Riedl, J. vom Brocke, P. M. Léger, and A. Randolph, Eds., Springer, Cham, vol. 16, 2017, pp. 157–163. https://doi.org/10.1007/978-3-319-41402-7_20
- [10] G. Da Costa, "Optical remote sensing of heartbeats," *Optics Communications*, vol. 117, nos. 5–6, pp. 395–398, 1995. [https://doi.org/10.1016/0030-4018\(95\)00181-7](https://doi.org/10.1016/0030-4018(95)00181-7)
- [11] G. Scebba, L. Tüshaus, and W. Karlen, "Multispectral camera fusion increases robustness of ROI detection for biosignal estimation with nearables in real-world scenarios," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 5672–5675. <https://doi.org/10.1109/EMBC.2018.8513501>
- [12] G. Scebba, G. Da Poian, and W. Karlen, "Multispectral video fusion for non-contact monitoring of respiratory rate and apnea," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 1, pp. 350–359, 2021. <https://doi.org/10.1109/TBME.2020.2993649>
- [13] S. Baker, W. Xiang, and I. Atkinson, "Determining respiratory rate from photoplethysmogram and electrocardiogram signals using respiratory quality indices and neural networks," *PLoS ONE*, vol. 16, no. 4, p. e0249843, 2021. <https://doi.org/10.1371/journal.pone.0249843>
- [14] A. Kwasniewska, J. Ruminski, and M. Szankin, "Improving accuracy of contactless respiratory rate estimation by enhancing thermal sequences with deep neural networks," *Appl. Sci.*, vol. 9, no. 20, p. 4405, 2019. <https://doi.org/10.3390/app9204405>
- [15] D. Bian, P. Mehta, and N. Selvaraj, "Respiratory rate estimation using PPG: A deep learning approach," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 5948–5952. <https://doi.org/10.1109/EMBC44109.2020.9176231>
- [16] H. Ghanadian and H. Al Osman, "Non-contact heart rate monitoring using multiple RGB cameras," in *Computer Analysis of Images and Patterns, CAIP 2019*, in Lecture Notes in Computer Science, M. Vento and G. Percannella, Eds., Springer, Cham, vol. 11679, 2019. https://doi.org/10.1007/978-3-030-29891-3_8
- [17] E. Moya-Albor, J. Brieva, H. Ponce, and L. Martínez-Villaseñor, "A non-contact heart rate estimation method using video magnification and neural networks," *IEEE Instrumentation & Measurement Magazine*, vol. 23, no. 4, pp. 56–62, 2020. <https://doi.org/10.1109/MIM.2020.9126072>

- [18] Z. El Khadiri, R. Latif, and A. Saddik, "Remote heart rate measurement using plethysmographic wave analysis," in *Advances in Machine Intelligence and Computer Science Applications, ICMICSA 2022*, in Lecture Notes in Networks and Systems, N. Aboutabit, M. Lazaar, and I. Hafidi, Eds., Springer, Cham, vol. 656, 2023, pp. 254–267. https://doi.org/10.1007/978-3-031-29313-9_23
- [19] Z. El Khadiri, R. Latif, and A. Saddik, "Breathing pattern assessment through the empirical mode decomposition and the empirical wavelet transform algorithms," in *The 3rd International Conference on Artificial Intelligence and Computer Vision (AICV2023)*, AICV 2023, in Lecture Notes on Data Engineering and Communications Technologies, A. E. Hassanien et al., Eds., Springer, Cham, vol. 164, 2023, pp. 262–271. https://doi.org/10.1007/978-3-031-27762-7_25
- [20] H. Yu, G. Zhang, J. Liu, and K. Li, "Intelligent knowledge service system based on depression monitoring of college students," *International Journal of Emerging Technologies in Learning (ijET)*, vol. 14, no. 12, pp. 71–84, 2019. <https://doi.org/10.3991/ijet.v14i12.10702>
- [21] Z. El Khadiri, R. Latif, and A. Saddik, "Efficient remote health monitoring using deep learning and parallel systems," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 10, 2024. <https://doi.org/10.14569/IJACSA.2024.0151020>
- [22] H. El Boussaki, R. Latif, and A. Saddik, "A review on the driver's fatigue detection methods," in *Digital Technologies and Applications, ICDTA 2023*, in Lecture Notes in Networks and Systems, S. Motahhir and B. Bossoufi, Eds., Springer, Cham, vol. 668, 2023, pp. 464–473. https://doi.org/10.1007/978-3-031-29857-8_47
- [23] H. El Boussaki, R. Latif, and A. Saddik, "A review on video-based heart rate, respiratory rate and blood pressure estimation," in *Advances in Machine Intelligence and Computer Science Applications, ICMICSA 2022*, in Lecture Notes in Networks and Systems, N. Aboutabit, M. Lazaar, and I. Hafidi, Eds., Springer, Cham, vol. 656, 2023, pp. 129–140. https://doi.org/10.1007/978-3-031-29313-9_12
- [24] Z. El Khadiri, R. Latif, and A. Saddik, "An efficient hybrid algorithm for non-contact physiological sign monitoring using plethysmography wave analysis," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 11, no. 6, pp. 2305–2321, 2023. <https://doi.org/10.1080/21681163.2023.2227741>
- [25] W. Jenkal et al., "An efficient method of ECG signals denoising based on an adaptive algorithm using mean filter and an adaptive dual threshold filter," *Int. Rev. Comput. Softw. (IRECOS)*, vol. 10, no. 11, pp. 1089–1095, 2015. <https://doi.org/10.15866/irecos.v10i11.7821>
- [26] H. El Boussaki, R. Latif, and A. Saddik, "Video-based heart rate estimation using embedded architectures," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023. <https://doi.org/10.14569/IJACSA.2023.01405119>
- [27] H. El Boussaki, R. Latif, A. Saddik, Z. El Khadiri, and H. El Boujaoui, "Non-contact respiratory rate monitoring based on the principal component analysis," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 9, 2023. <https://doi.org/10.14569/IJACSA.2023.01409106>
- [28] W. Jenkal, R. Latif, A. Toumanari, A. Dliou, O. El B'charri, and F. M. R. Maoulainine, "An efficient algorithm of ECG signal denoising using the adaptive dual threshold filter and the discrete wavelet transform," *Biocybernetics and Biomedical Engineering*, vol. 36, no. 3, pp. 499–508, 2016. <https://doi.org/10.1016/j.bbe.2016.04.001>
- [29] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019. <https://doi.org/10.1109/JPROC.2019.2915983>

- [30] S. M. Hussain, K. Mohamad Yusof, S. Ashfaq Hussain, R. Asuncion, and S. Ghouse, "Integration of 4G LTE and DSRC (IEEE 802.11p) for enhancing vehicular network performance in IoV using optimal cluster-based data forwarding (OCDF) protocol," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 15, no. 14, pp. 111–124, 2021. <https://doi.org/10.3991/ijim.v15i14.19201>
- [31] H. Mahdi, B. Al-Bander, M. H. Alwan, M. S. Abood, and M. M. Hamdi, "Vehicular networks performance evaluation based on downlink scheduling algorithms for high-speed long term evolution – vehicle," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 15, no. 21, pp. 52–66, 2021. <https://doi.org/10.3991/ijim.v15i21.22475>
- [32] S. M. Nasution, E. Husni, K. Kuspriyanto, R. Yusuf, and R. Mulyawan, "Road information collector using smartphone for measuring road width based on object and lane detection," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 14, no. 2, pp. 42–61, 2020. <https://doi.org/10.3991/ijim.v14i02.11530>
- [33] M. V. Yashina, A. I. Mokhov, M. A. Belova, A. V. Kostsov, and P. I. Pospelov, "On-board video-evaluation algorithm of transverse safety clearance for ahead road-vehicle," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 14, no. 10, pp. 128–138, 2020. <https://doi.org/10.3991/ijim.v14i10.14619>
- [34] H. Fechtner, K. H. Saes, E. Fechtner, T. Braun, and B. Schmülling, "Clarification of the training requirements for working on electric vehicles," *International Journal of Advanced Corporate Learning (ijAC)*, vol. 9, no. 1, pp. 35–43, 2016. <https://doi.org/10.3991/ijac.v9i1.5635>
- [35] A. Podlubne and D. Göhringer, "A survey on adaptive computing in robotics: Modelling, methods and applications," *IEEE Access*, pp. 53830–53849, 2023. <https://doi.org/10.1109/ACCESS.2023.3281190>
- [36] F. Zouari, "Neural network based adaptive backstepping dynamic surface control of drug dosage regimens in cancer treatment," *Neurocomputing*, vol. 366, pp. 248–263, 2019. <https://doi.org/10.1016/j.neucom.2019.07.096>
- [37] F. Salvetti, R. Gardner, R. Minehart, and B. Bertagni, "Voice- and Touch-controlled checklists," *International Journal of Advanced Corporate Learning (ijAC)*, vol. 13, no. 2, pp. 67–76, 2020. <https://doi.org/10.3991/ijac.v13i2.17037>
- [38] M. Tezer, N. A. Orekhovskaya, E. F. Shaleeva, S. A. Knyazeva, and J. A. Krokhina, "The effectiveness of STEM education applied with a distance education approach," *International Journal of Emerging Technologies in Learning (ijET)*, vol. 16, no. 19, pp. 180–192, 2021. <https://doi.org/10.3991/ijet.v16i19.26061>
- [39] V. Shurygin, A. Berestova, T. Litvinova, E. Kolpak, and A. Nureyeva, "Universal models and platforms in e-learning," *International Journal of Emerging Technologies in Learning (ijET)*, vol. 16, no. 9, pp. 63–75, 2021. <https://doi.org/10.3991/ijet.v16i09.19697>
- [40] A. Ibeas, A. Esmaili, J. Herrera, and F. Zouari, "Discrete-time observer-based state feedback control of heart rate during treadmill exercise," in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, 2016, pp. 537–542. <https://doi.org/10.1109/ICSTCC.2016.7790721>
- [41] R. Ranjan, J. L. López, K. Lal, S. Saxena, and S. Ranjan, "Adopting a new hybrid force model: A survey during Covid-19 in indian higher education," *International Journal of Emerging Technologies in Learning (ijET)*, vol. 16, no. 16, pp. 169–185, 2021. <https://doi.org/10.3991/ijet.v16i16.23371>
- [42] J. E. Stone, D. Gohara, and G. Shi, "OpenCL: A parallel programming standard for heterogeneous computing systems," *Computing in Science & Engineering*, vol. 12, no. 3, pp. 66–73, 2010. <https://doi.org/10.1109/MCSE.2010.69>
- [43] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004. <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>

- [44] D. Cochran, "A consequence of signal normalization in spectrum analysis," in *ICASSP-88, International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 1988, pp. 2388–2389. <https://doi.org/10.1109/ICASSP.1988.197121>
- [45] M. P. Tarvainen, P. O. Ranta-Aho, and P. A. Karjalainen, "An advanced detrending method with application to HRV analysis," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 2, pp. 172–175, 2002. <https://doi.org/10.1109/10.979357>
- [46] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Phil. Trans. R. Soc. A*, vol. 374, pp. 1–16, 2016. <https://doi.org/10.1098/rsta.2015.0202>
- [47] D. P. Rodgers, "Improvements in multiprocessor system design," *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3, pp. 225–231, 1985. <https://doi.org/10.1145/327070.327215>
- [48] Reddy, Martin, *API Design for C++*. Burlington, Massachusetts: Morgan Kaufmann Publishers, 2011.
- [49] X. Guo, J. Wu, Z. Wu, and B. Huang, "Parallel computation of aerial target reflection of background infrared radiation: Performance comparison of OpenMP, OpenACC, and CUDA implementations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 4, pp. 1653–1662, 2016. <https://doi.org/10.1109/JSTARS.2016.2516503>
- [50] J. Diaz, C. Muñoz-Caro, and A. Niño, "A survey of parallel programming models and tools in the multi and many-core era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1369–1386, 2012. <https://doi.org/10.1109/TPDS.2011.308>
- [51] J. Wolfer, "Embedding topical elements of parallel programming, computer graphics, and artificial intelligence across the undergraduate CS required courses," *International Journal of Engineering Pedagogy (IJEP)*, vol. 5, no. 1, pp. 27–32, 2015. <https://doi.org/10.3991/ijep.v5i1.4090>
- [52] L. G. García, E. Montoya, S. Isaza, and R. A. Velasquez, "An open edX extension for parallel programming assignments with automatic configurable grading," *International Journal of Engineering Pedagogy (IJEP)*, vol. 11, no. 4, pp. 7–22, 2021. <https://doi.org/10.3991/ijep.v11i4.20097>
- [53] K. Orynbekova, A. Bogdanchikov, S. Cankurt, A. Adamov, and S. Kadyrov, "MapReduce solutions classification by their implementation," *International Journal of Engineering Pedagogy (IJEP)*, vol. 13, no. 5, pp. 58–71, 2023. <https://doi.org/10.3991/ijep.v13i5.38867>

8 AUTHORS

Zakaria El Khadiri received his M.S. degree from Ibn Zohr University, Morocco in 2020. Currently, he is a PhD student at the Laboratory of Systems Engineering and Information Technology (LISTI) laboratory ENSA, University Ibn Zohr, Agadir, Morocco. His current research focuses on computer science and embedded systems (E-mail: zakaria.elkhadiri@edu.uiz.ac.ma).

Rachid Latif received a PhD. degree in signal processing in 2000, a Habilitation degree in 2005 from the faculty of sciences, and a full professor in Information Technology in 2015, from Ibn Zohr University, Morocco. Currently, he is a Professor at the Department of Industrial & Electrical Engineering, National School of Applied Sciences, Ibn Zohr University, Agadir, Morocco. His research interests include Biomedical signal and image processing, Biomedical instrumentation, Robotics and Embedded systems (CPU, DSP, FPGA, CPU-GPU, CPU-FPGA), and Artificial intelligence and he is working on the modeling, filtering, and analysis of fetal cardiac signals and monitoring of premature babies. From 2006 to 2014, he was the head of the signals systems and computer science team (ESSI). He is the head of the LiSTi Laboratory.

Besides, he is the head of the Moroccan Society of Embedded Systems (MSES) and the head of Embedded Systems and Biomedical Engineering Master (SEIB), and also, he is also an evaluator expert at CNRST Morocco.

Amine Saddik received his M.S. degree from Ibn Zohr University, Morocco, in 2019, and his Ph.D. degree in Computer Science and Embedded Electronics Systems from Ibn Zohr University in collaboration with the University of Paris-Saclay, France, in 2022. He is currently an Associate Professor at the Faculty of Applied Science, Ibn Zohr University, Morocco.