|  |  |
| --- | --- |
| **ITU – Telecommunications Standardization Sector**  STUDY GROUP 21 Question C/16  **Video Coding Experts Group (VCEG)**  75th Meeting: 2-8 November 2024, Kemer, TR | Document VCEG-BW04-v1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Question: | C/16 SG21 (VCEG) | | |
| Source: | **Ray Krasinski, Sam Jelfs (Philips)** | Email: | [Raymond.Krasinski@philips.com](mailto:Raymond.Krasinski@philips.com) |
| Title: | **Description of Philips’s response to the Call for Proposals on the compression of biomedical waveform data** | | |
| Purpose: | Proposal | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

Philips is a diversified technology company that develops and manufactures medical systems and solutions for sale throughout the world. Philips offers products and solutions in the areas of diagnostic imaging, enterprise diagnostic informatics, image-guided therapy, ultrasound, monitoring and analytics, sleep and respiratory care, population health management, connected care informatics, and therapeutic care. Philips has a long history of contributing to the standardization of technologies to improve data and device interoperability with the ultimate goal of improving people’s health and well-being.

This document describes Philips’s response to the Call for Proposals (CfP) on the coding of biomedical waveform data (provisional name H.BWC) [1].

As a result of unforeseen, last minute procedural issues with the selected EEG data set in the CfP, there was no time left to get internal Philips approval for an appropriate EEG alternative data set in place. Therefore, this Philips CfP submission is limited to the ECG and EMG datasets. It is noted that the CfP does not mandate submission to cover all three datasets.

Section 2 of this document outlines the considerations into preparing the submission.

Section 3 comprises a technical description of the proposed technology. The technology is largely based on Wavelet transform. Wavelet transforms, and in particular the Discrete Wavelet Transform (DWT), are well suited for processing of signals of physical nature (time domain, discrete type) that can be found in real-life. Consequently, it is therefore well established to also use DWTs for processing of biomedical (physiological) signals [2].

As stipulated in the CfP, the proponent has submitted bitstreams and decoder executables in advance of this contribution describing the submitted technology. The bitstreams cover all input sequences and working points for each of the two categories:

* Electrocardiography (ECG) signals
* Electromyography (EMG) signals

It is noted that the collection of EEG recordings from patients at the Medical University of South Carolina has been removed.

Section 4 describes the runtime environment for decoding the bitstreams and storing the decoded data in the same format as the test sequence format. The software that is provided is capable of both encoding and decoding the bitstreams, and is written in the Python language which has been compiled into an executable for submission.

Section 5 describes the decoding complexity characteristics for implementation of the proposed technology (in terms of memory capacity for programs and fixed data tables, working data storage, computational requirements, etc.) as well as information about the encoding algorithm used to generate the submitted bitstreams.

Section 6 provides the average PRD, CPRD, and PSNR values for each of the submitted bitstreams along with the number of bits per sample (BPS).

# Considerations

Since, different from e.g., Audio and Video compression, biomedical signals are subject to clinical subjective evaluation, often after feature processing, it is of the utmost importance that clinical features remain intact. The proponents are confident that the success of H.BWC will largely be determined by these aspects. In fact, several Philips businesses have resorted to developing their own (point) solution for compression of a biomedical signal (specifically ECG) so that relevant clinical artefacts remain intact.

Philips businesses manufacture and sell a full line of hospital cardiographs, clinical cardiology Holter monitoring, wearable solutions, and stress testing devices in addition to selling an ECG Management System and analytic algorithms to support the analysis of collected cardiology data. Philips manufactures and sells an innovative “beltless” fetal monitor solution that can measure ECG and EMG signals in both the fetus and the mother. These devices and solutions are used both within a hospital or clinic setting as well as for outpatient use cases. The collection, management and analysis of the data collected is an integral part of these solutions.

As the diagnosis and treatment of healthcare patients slowly transitions from a “one size fits all” approach of medical care that has been in use for the past 100 years, towards evidence-based treatment optimized for the specific characteristics of the patient, the data needs of the industry is changing. This more personalized approach to diagnosis and treatment is known as Precision Medicine. In order for Precision Medicine to be effective, more data is needed about each individual patient, both a larger variety of data types as well as a larger volume of each specific data type. Genomic sequences are an example of very large newer data sets that did not exist in the clinical space 25 years ago but are now being applied to diagnosis and can be hundreds of gigabytes in size. This is not the only driver of increased data volume however. Where in the past short duration time series data samples, measured in seconds, would be collected, such as heart rate signals, today clinicians collect long time duration data sequences, spanning many days. These longer duration time series data sets provide a more complete picture of the patient but require exponentially more data storage.

More advanced clinicians and institutions are examining long time duration data across different data types (modalities) to search for correlations and to improve diagnosis and, ultimately, improve treatments. The ability to store, manage and make use of these large clinical data sets, both individually and in combination, is increasingly necessary but unsupported by existing data encoding methods.

In addition to the clinical domain, the consumer health, wellness and fitness space is also generating significant amounts of data. With the increasing popularity of wearable devices (such as sports watches), data capture in the pro/ consumer space (such as phone-camera, automotive-camera capture of vital signs), and the inevitable shift to home patient monitoring and 24/7 wireless monitoring in and outside of the hospital, the need for a universal interoperable solution for biomedical signals becomes more relevant than ever. Consumer wearable fitness data has been slowly infiltrating the clinical domain and this trend is expected to accelerate.

Much of the data being collected is analyzed, at least in part, through automated signal processing techniques, either classical DSP or using machine learning / AI, yet the “diagnosability” [3] of the signals is also critical. This is the value to, and ability for, clinicians to be able to diagnose symptoms from visual inspection of the signal. Typically, when we analyze a compression method, the Percentage RMS Difference (PRD) is used as an error metric, yet this is known to be a poor estimate of diagnosability. At least for ECG analysis an error metric such as the Weighted Diagnostic Distortion (WDD) [4] can be used. By example, Figure 1 shows three manipulations of an ECG signal through either low-pass filtering at 40Hz, reduction of the R peak, or flattening of the Q and S waves. In each case the PRD with respect to the original signal is ~11.5, yet especially in the case of the Q-S wave removal the diagnostic value of the signal is significantly changed. In many well-known ECG analysis methods, such as the Pan-Tompkins algorithm for R peak detection [5], the first stage is band-pass filtering to increase the signal-to-noise ratio. As such, perseverance of the high-frequency “noise” component should not be traded for ensuring that diagnostic features are maintained.

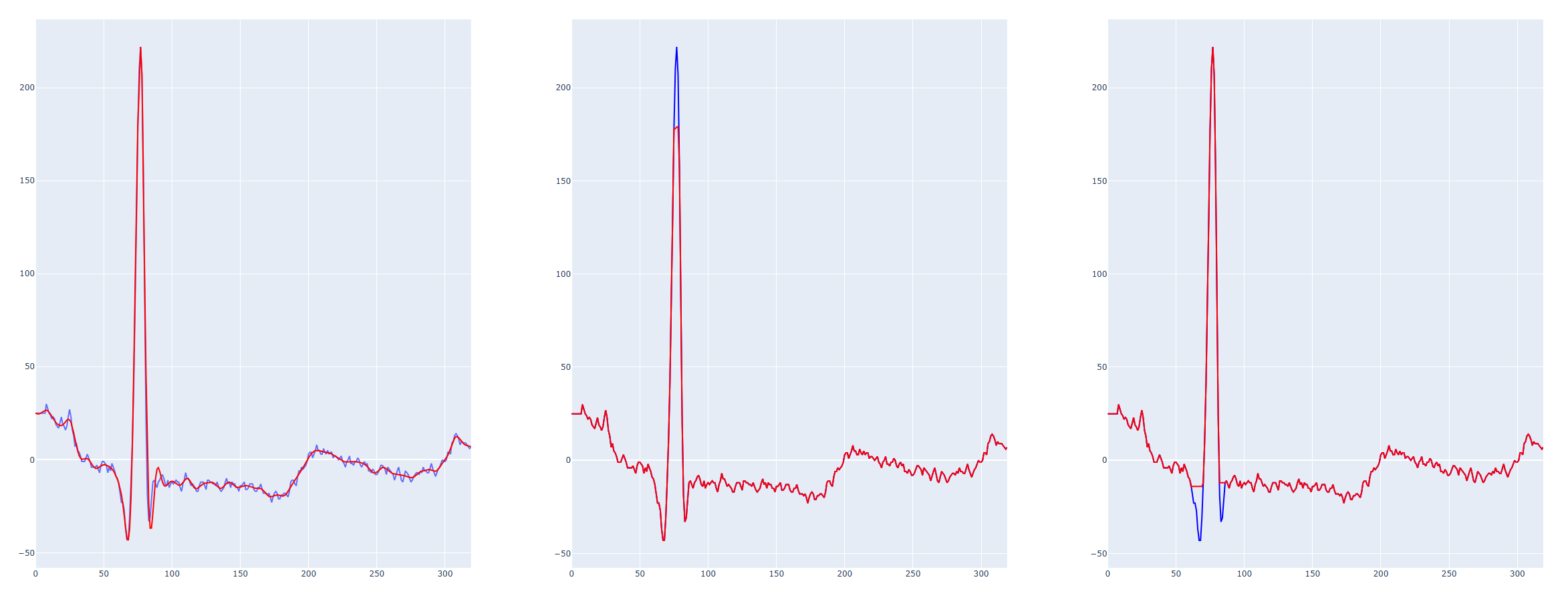


Figure 1 - Comparison of 3 ECG distortions, all having the same PRD (~11.5). Left - Low-pass filter at 40Hz, Center - Flattening of R Peak, Right, - Flattening of Q and S waves. Blue trace shows the original signal in all plots.

These developments demonstrate that

* next to ECG, EEG and EMG signals, support for Photoplethysmography (PPG) signals as these are mainly used in wearables, will be an important and logical addition to this standardization activity.
* modern datasets are multimodal and combine various combinations of biomedical and even non-biomedical signals that in some cases may be used for further postprocessing, such as feature extraction. For example, ECG and PPG signals are typically accompanied by accelerometer and skin/ air temperature signals.
* the need is for accommodating multiple modalities, rather than a high number of channels of a specific modality, such as e.g., in EEG signals. For the latter, dedicated solutions are well established within their eco systems and unlikely to move to a new format.
* there is clear benefit in having a single format that is able to capture different modalities as well as their interrelation. For example, patient monitoring systems and ambulatory monitors monitor multiple modalities in a synchronized fashion allowing the healthcare professional to overlay and apply sophisticated multimodal feature analyses.
* In many cases, biomedical signals are to be transmitted over mobile communication channels (another area fitting the ITU-T scope), putting constraints on the resources available.
* The diagnostic features of the signal should be maintained

A universal compression solution that takes these aspects into account is therefore of high relevance, both in the consumer and professional market, but also in the area between.

# Technical description of the codec architecture

## High level description

The proposed compression algorithm partitions the biomedical input signals into blocks. The block size is flexible such that, depending on the use case, the appropriate trade-off between compression efficiency and random access can be achieved. The proposal is based upon straightforward, well-established technologies. Depending on the compression target, one of two coding methods is applied:

1. Wavelet transform in combination with Huffman coding and optionally an additional lossless compression stage for high compression (lower bitrates)
2. JPEG2000 [6] style compression (which inherently employs Wavelet transform) for low compression up to lossless compression (higher bitrates)

Similar to JPEG2000, rather than specifying a target bitrate, a numeric compression level is specified at the encoder. This is a value from 0 to 9 that corresponds with lossless to maximal compression.

In addition, the algorithm further supports the following features:

* DC coding.
* Adaptive grouping over channels, modalities and over time
* Annotation data

In the subsequent subsections, the two coding methods and additional features are described in more detail.

## Wavelet Transform based coding stage

Figure 2 illustrates the Wavelet Transform based coding stage that is employed for the high compression (low bitrate). It comprises of a DWT Wavelet transform which coefficients are quantized and entropy coded. Optionally an additional lossless coding stage is supported for increased compression performance.



Figure 2 - Wavelet Transform based coding stage

As illustrated in Figure 3, DWT splits the time-domain signal into a low frequency band using a filter g0[n] and a high frequency band using a filter h0[n]. The low-band filter g0[n] and the high-band filter h0[n] are complementary filters such that these constitute a critically sampled filter-bank. I.e., the outputs of the analysis filters g0[n] and h0[n] may be critically sampled (decimated by a factor 2) such that the sampling rate at the input time-domain signal x[n] is the same as the (combined) sampling rate at the outputs of the analysis filters g0[n] and h0[n]. There are thus no additional samples (data) generated at the output of the analysis filterbank. Using a complementary synthesis structure with filters g1[n] and h1[n], perfect reconstruction can be obtained. The choice of the wavelet (the ‘Mother Wavelet’) provides a degree of freedom in the design of the DWT.

A 'Mother Wavelet' refers to a polyharmonic wavelet function that serves as a basis for generating other wavelets through scaling and translation operations. The term ‘Mother wavelet’ is used to describe its role as the foundational function for generating a complete set of wavelets.

Critical sampling of filterbanks as well as (near) perfect reconstruction is well known from the audio coding domain using critically sampled filterbanks such as Quadrature Modulate Filter (QMF) or Modulated Discrete Cosine Transform (MDCT) filter banks. Typically, the filterbanks used in the audio coding domain use much more than 2 filters, for example 32 or 64.

As illustrated in Figure 3, in a DWT, for each subsequent level, only the low band is further split using the same analysis filter g0[n], resulting in a further split of the low frequency band until eventually a filter tree up to a level J is obtained.



Figure 3 - DWT is a concatenation of a two-band splitters. DWT splits the time-domain signal into a low frequency band using a filter g0[n] and a high frequency band using a filter h0[n]

As illustrated in Figure 4, the LF portions are denoted as 'Approximation coefficients'​. The HF portions are denoted as 'Detail coefficients'​. Detail coefficients can be seen as what is missing from the approximation at level J to get to the​ approximation at level J-1.



Figure 4 - The LF portions are denoted as 'Approximation coefficients'​. The HF portions are denoted as 'Detail coefficients'​.

When applying a DWT onto a time domain signal, it produces a set of coefficients that represent specific time-frequency tiles as illustrated in Figure 5, where each block represents a specific time-frequency interval, i.e., an area bounded by a specific time and frequency interval. In practice, the ‘bounding’ is not hard because of the non-perfect resolution of practical filters (as can also been seen at the overlap regions in Figure 3). Low frequency coefficients have a high frequency resolution but a low time resolution​. On the other hand, high-frequency coefficients have a low frequency resolution but high time resolution (see Heisenberg boxes/rectangles in time-frequency representations). This combination of frequency and time resolution is known to match well to physical processes. In comparison, general subband- or frequency-transform filterbanks employ a uniform division of the time/frequency plane.



Figure 5 - Wavelet Time-Frequency coefficients as a function of the level of the DWT

The DWT is applied to subsequent segments of a signal so that a block-based processing is enabled, which determines the random access. For each channel, wavelet coefficients are organized into an array of approximation coefficients followed by the detail coefficients from low to high frequency. For coding multiple channels, wavelet coefficients are interleaved. Next, the coefficients are scaled using a scalefactor that is determined by a.o., the compression level and quantized using a fixed quantizer. Finally, the thus obtained (integer) quantization levels are efficiently entropy coded by a combination of run-length coding of leading zeros and Huffmann tables. Tables are optimized for signal type and compression level.

Optionally, an additional lossless compression step (standard zlib) is supported to further increase the coding efficiency.

## JPEG2000 based coding

For lower compression ratios (higher bitrates), standard ITU-T JPEG2000 compression is employed. The time domain channels for each channel are organized into a 2-D (image) matrix, row by row. A compression level of 0...9 can be set, where 0 represents lossless and 9 highest compression. The aspect ratio is determined in the encoder, primarily as a function of signal type, sampling rate and block length.

## DC coding

For some signal types such as ECG, retaining the DC component is very important as specific features such as breathing rate are derived from the DC component. The DC component is separately encoded on a frame basis.

## Adaptive grouping over channels, modalities and over time

The submission accommodates adaptive grouping over channels and modalities over specific time segments. So, for specific periods of time, specific combinations of channels/ modalities may be coded together thereby exploiting redundancy between the channels. Since, in many cases, channels a/o signals of multiple modalities may not be aligned and the alignment may even change over time, synchronization is supported. This not only benefits the joint coding, but also any multi-modal clinical evaluation/ feature extraction of the decoded signals as they will be properly aligned.

## Annotation data

The proposalsupports inclusion of annotation meta data. These metadata may be relevant for subsequent clinical evaluation/ feature extraction.

# Bitstream Encoding and Decoding

The provided software is capable of encoding bitstreams from EDF files. The basic process is to read the raw digital signals from the EDF file, segment it into frames of the desired length, and depending on the signal modality and the desired compression level, either carry out the wavelet coding as defined in 3.2 or the JPEG coding as defined in 3.3. The bitstream syntax for decoding is provided below, and, as such, the encoding should be easily understood.

Bitstream decoding follows a basic principle, firstly a header is read that contains information related to the contained signals, followed by independent data frames. The data frames may contain either JPEG2000 or coded wavelet coefficients, depending on the signal type and desired compression level.

The syntax of the header is given in Table 1. For the sake of brevity, syntax for reading patient data and grouping information is not included. Table 2 and Table 3 contain the syntax for reading compressed wavelet coefficients and JPEG2000 data respectively.

Table 1 - Syntax of bitstream header

|  |  |  |
| --- | --- | --- |
| Syntax | No. of bits | Mnemonic |
| compressedMedicalSignalHeader() |  |  |
| { |  |  |
| **versionNumber**; | **8** | **uimsbf** |
| **hasPatientData**; | **1** | **bslbf** |
| If (hasPatientData) { |  |  |
| readPatientData() |  |  |
| } |  |  |
| maxNumSamplesPerFrame | **20** | **uimsbf** |
| startTime | **32** | **simsbf** |
| **numChannels** | **8** | **Uimsbf** |
| **hasGrouping;** | **1** | **Bslbf** |
| If (hasGrouping) { |  |  |
| readGroupInfo() |  |  |
| } else { |  |  |
| **globalCompressionLevel** | **4** | **Uimsbf** |
| **globalSampleRate** | **4** | **Uimsbf** |
| **globalSignalType** | **4** | **Uimsbf** |
| **globalDigitalRange** | **4** | **uimsbf** |
| if (globalDigitalRange == “custom”) { |  |  |
| **globalDigitalMin** | **20** | **simsbf** |
| **globalDigitalMax** | **20** | **simsbf** |
| } |  |  |
| **globalAnalogueUnits** | **4** | **uimsbf** |
| **globalAnalogueRange** | **4** | **uimsbf** |
| if globalAnalogueRange == “custom”) { |  |  |
| **globalAnalogueMin** | **20** | **simsbf** |
| **globalAnalogueMin** | **20** | **simsbf** |
| } |  |  |
| } |  |  |
|  |  |  |
| for (int I = 0; I < numChannels; i++) { |  |  |
| startingDCOffset | **20** | **simsbf** |
| } |  |  |
|  |  |  |
| **hasAnnotations;** | **1** | **bslbf** |
| byteAlign() |  |  |
| } |  |  |

Table 2 - Syntax to read frame for wavelet compressed data

|  |  |  |
| --- | --- | --- |
| Syntax | No. of bits | Mnemonic |
| readFrame() |  |  |
| { |  |  |
| if (maxNumSamplesPerFrame > ThresholdForZipCompression) { |  |  |
| **lengthCompressedFrame** | 24 | uimsbf |
| **compressedData** | lenghCompressedFrame \* 8 | binary data |
| decompressedData = zlib.decompress(compressedData)[zeroPadding:end] |  |  |
| } |  |  |
| If (**hasCustomNumberOfBitsForOffset**){ | 1 | bslbf |
| **numBitsForOffset** | 5 | uimsbf |
| } else { |  |  |
| numBitsForOffset = DefaultBitsForOffset[quantization] |  |  |
| } |  |  |
| sampleIdx = -1 |  |  |
| while (sampleIdx < maxNumberSamplesPerFrame){ |  |  |
| if (**hasSampleOffset**) { | **1** | **bslbf** |
| sampleIdx += **sampleOffset** | numBitsForOffset | **uimsbf** |
| } else ( |  |  |
| sampleIdx += 1 |  |  |
| } |  |  |
| for (int i = 0; i < numChannels; i++) { |  |  |
| if (**hasCoeff**){ | **1** | **bslbf** |
| signal[i][sampleIdx] = readCoeff() \* quantization[i] |  |  |
| } |  |  |
| } |  |  |
| } |  |  |
| for (int i = 0; i < numChannels; i++) { |  |  |
| if (**hasEndDCOffset**){ |  |  |
| **dcOffset[i]** | **4** | **uimsbf** |
| } |  |  |
| } |  |  |
| } |  |  |

Table 3 - Syntax to read frame of JPEG2000 compressed data

|  |  |  |
| --- | --- | --- |
| Syntax | No. of bits | Mnemonic |
| readFrameJPEG() |  |  |
| { |  |  |
| for (int i = 0; i < numChannels; i++){ |  |  |
| **len\_image\_data** | **32** | **uimsbf** |
| **compressedData** | **len\_image\_data** | **binary data** |
| decompressedData = JPEG2000.Decompress(compressedData) |  |  |
| signal[i] = reshape(decompressedData,maxNumSamplesPerFrame) |  |  |
| } |  |  |
| for (int i = 0; i < numChannels; i++) { |  |  |
| if (**hasEndDCOffset**){ |  |  |
| **dcOffset[i]** | **4** | **uimsbf** |
| } |  |  |
| } |  |  |
| } |  |  |

Wavelet coefficient values are read using the syntax of Table 4 using the Huffman codes given in a look-up table that is stored in ROM.

Table 4 - Syntax to decode a wavelet coefficient

|  |  |  |
| --- | --- | --- |
| Syntax | No. of bits | Mnemonic |
| readCoeff() |  |  |
| { |  |  |
| val = LUT(coeffValues) | **var** | **vlclbf** |
| if (**isLargerNumber**){ | **1** | **bslbf** |
| val += LUT(coeffValues) \* 64 | var | vlclbf |
| if (**isEvenLargerNumer**){ | **1** | bslbf |
| val += LUT(coeffValues) \* 2048 | **var** | vlclbf |
| } |  |  |
| if (**isNegative**){ | 1 | bslbf |
| return -val |  |  |
| } else { |  |  |
| return val |  |  |
| } |  |  |

# Complexity

This section examines the compression algorithm’s decoding complexity, covering computational demands, memory requirements, and storage needs. The observed time complexities highlight the algorithm’s suitability for efficient biomedical waveform analysis, particularly in applications requiring high performance. It should be noted that the provided software, being compiled Python code, is not optimized and as such the algorithmic complexity should not be estimated from the implementation efficiency.

**Computational Complexity**:

Both the CDF9/7 algorithm using the lifting method that is used for the wavelet forward and inverse transforms and the JPEG2000 en-/de-coding are complexity, with respect to frame size. This efficiency makes both particularly advantageous for large datasets, where rapid, frame-based decompression is required for responsive performance. Efficient implementations of both CDF9/7 and JPEG2000 are readily available in various programming languages for easy integration. The reading of the wavelet coefficients and reconstruction of the signals from either wavelet or JPEG frames is theoretically .

**Memory Requirements**:

To optimize decoding, the algorithm uses Lookup Tables (LUTs) stored in ROM to hold predefined quantization levels and wavelet coefficients for ECG and EMG signals. These LUTs reduce runtime computation by enabling quick access to binary-coded values used in both encoding and decoding processes. Specifically, the quantization\_codes table stores compression levels in binary format, providing fast access to appropriate settings. Additionally, separate LUTs for ECG (wavelet\_coefficient\_codes\_ECG) and EMG (wavelet\_coefficient\_codes\_EMG) contain binary representations of the variable length wavelet coefficients tailored to each signal type. For instance, an ECG coefficient of 1 maps to "01", while EMG uses "00". These mappings streamline coefficient retrieval, ensuring decoding is efficient and consistent. Storing these mappings in ROM minimizes memory demands and provides fast, reliable access to essential data without recalculation. Each coefficient requires a maximum of 10 bits to store the value, and 4 bits to store the length so as to have the correct leading zero padding. Tables for common signal types may be included based on the implementation requirements, and a generic table is available for any unknown signal types.

**RAM Usage**:

For decompression processes, RAM requirements scale directly with the frame size. During JPEG decompression, each signal channel requires a dedicated memory buffer that matches the dimensions of the original frame to store decompressed pixel data independently. Similarly, wavelet decompression allocates memory per channel based on frame size, with each buffer holding decompressed wavelet coefficients. This channel-specific memory allocation supports concurrent processing, optimizing RAM utilization in complex, multi-channel datasets.

Overall, the compression algorithm’s decoding complexity reflects a carefully balanced approach to computational load and memory management, making it a robust choice for high-efficiency biomedical waveform processing.

# Experimental Results

Subsections 6.1 and 6.2 show graphs of PRD, CPRD, and PSNR values over BPS values for the ECG and EMG datasets respectively. Due to the late availability of the replacement EEG dataset it was not possible to process this data before the CfP submission deadline. In all plots, vertical dashed lines indicate the upper bounds of the operating points as indicated in [1]. As noted in 3.1, the encoder has predetermined operating points similar to those used in JPEG2000, for target BPS levels in between these provided operating points linear interpolation should be used.

The solution supports encoding and encoding the signals with variable frame durations. There is a trade-off between frame duration, compression ratio, memory demands, and random access in time. As no frame duration was specified in the CfP [1] a duration of 1 minute has been chosen for these results, but it should be noted that increasing the frame duration will naturally decrease the bitrate. Figure 12 shows an example of this trade-off for the ECG dataset at 3 different frame durations.

Full per-sequence results as well as a decoder executable and all bit streams have been uploaded to the FTP server specified in [1] as files VCEG-BT07-v1-Philips-response-results.zip and VCEG-BT07-v1-Philips-response.zip. Included with the response are bash scripts to facilitate the encoding and decoding of the CfP datasets.

## ECG

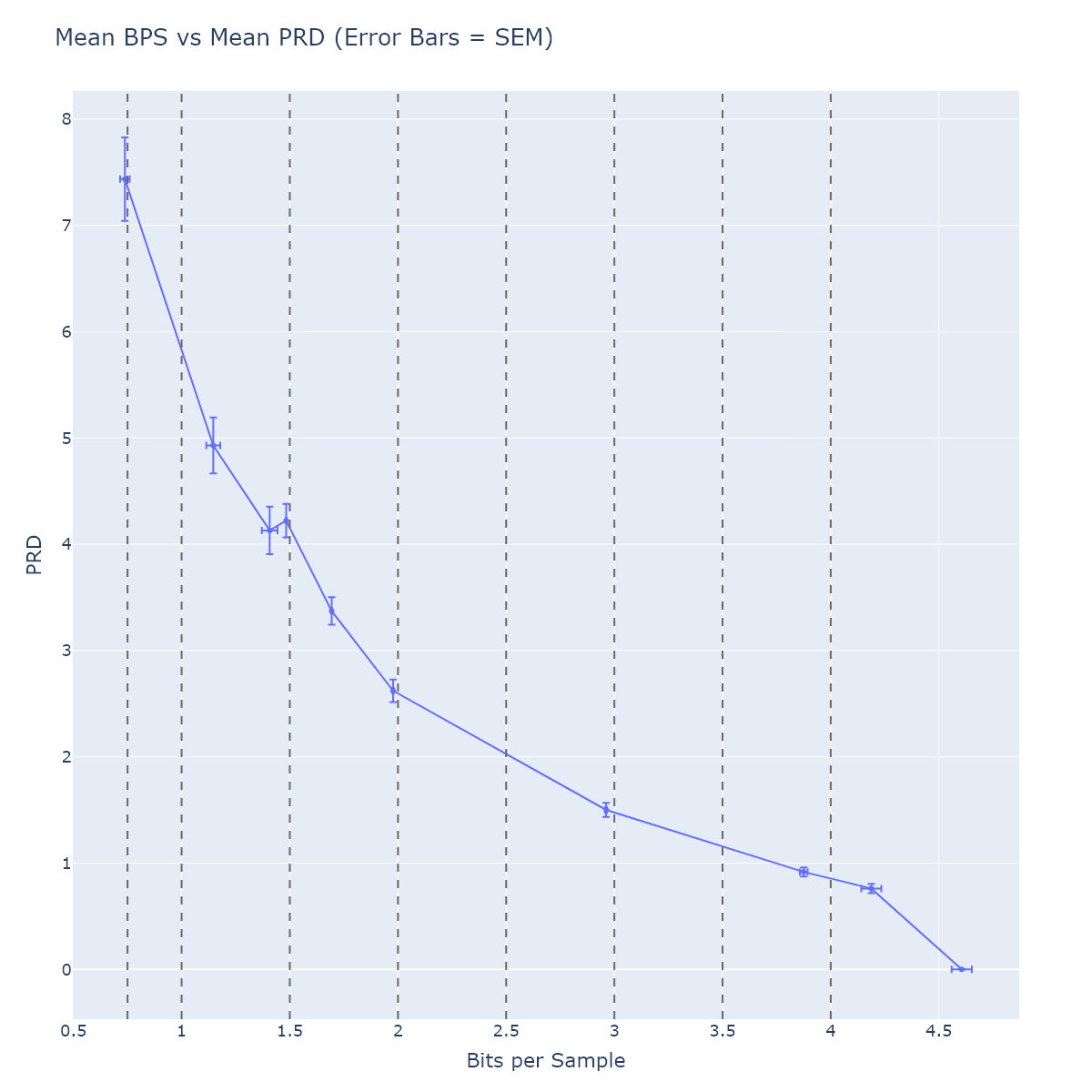


Figure 6 - Mean PRD values vs mean BPS per compression level. Error bars indicate standard error of the mean.

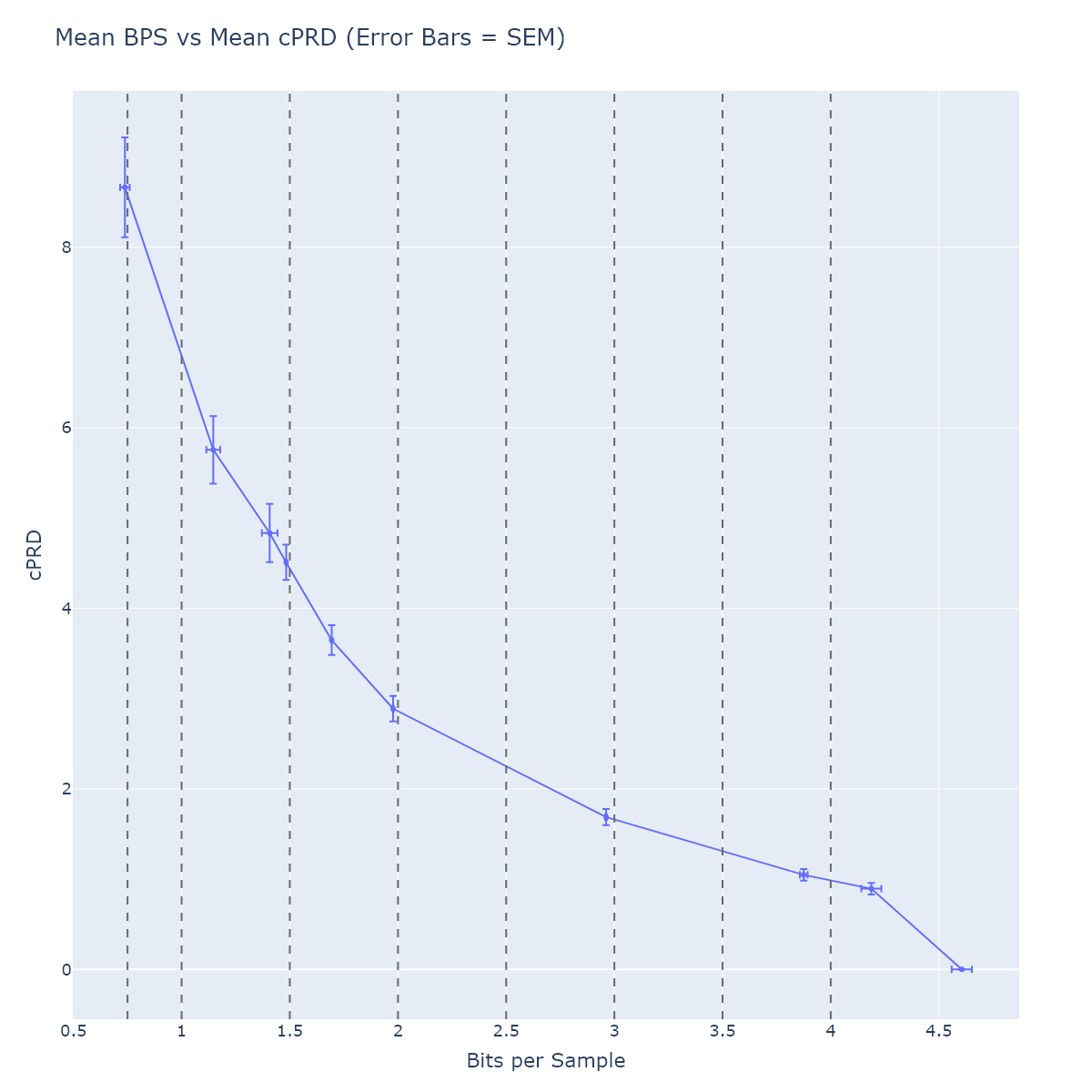


Figure 7 - Mean CPRD values vs mean BPS per compression level. Error bars indicate standard error of the mean.

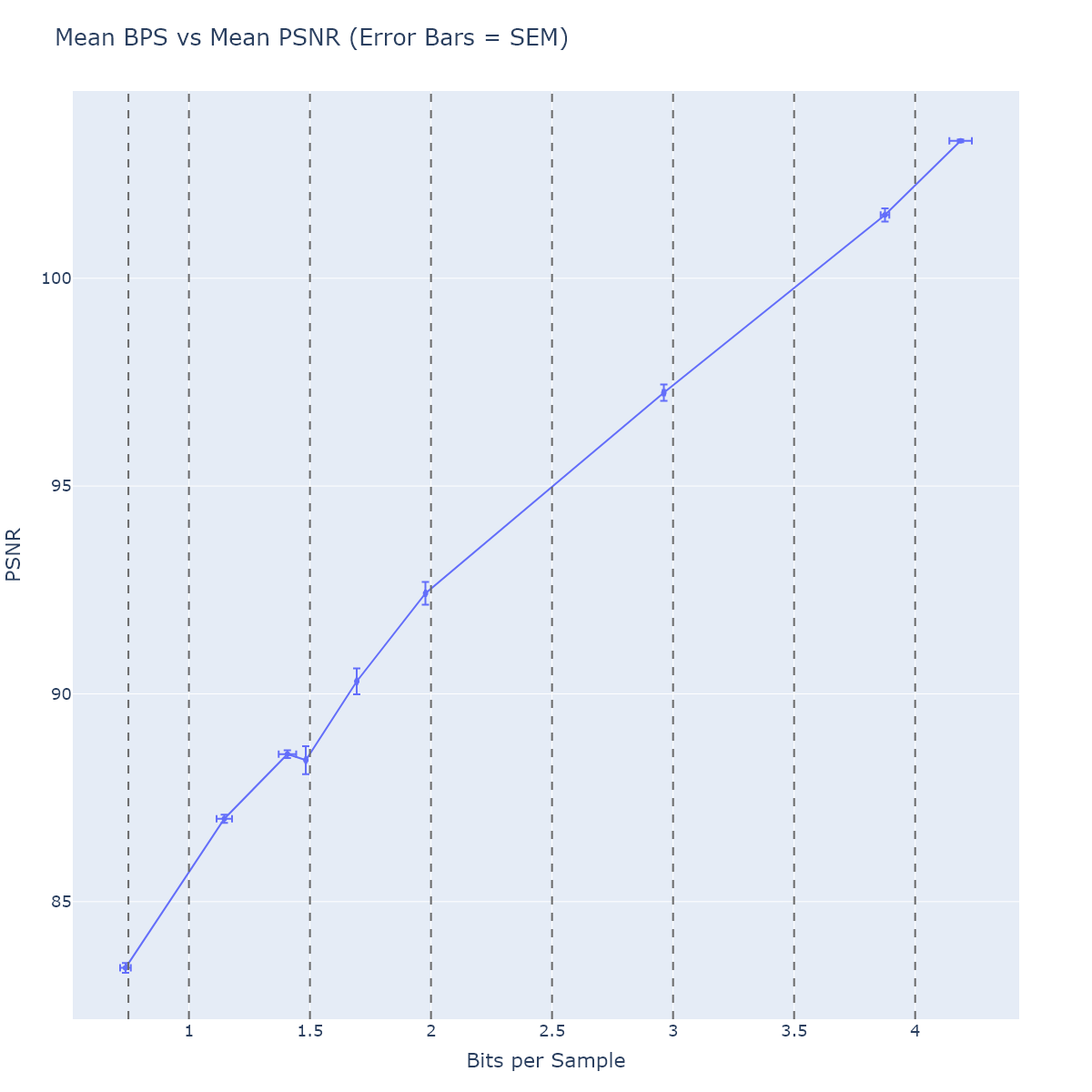


Figure 8 - Mean PSNR values vs mean BPS per compression level. Error bars indicate standard error of the mean.

## EMG

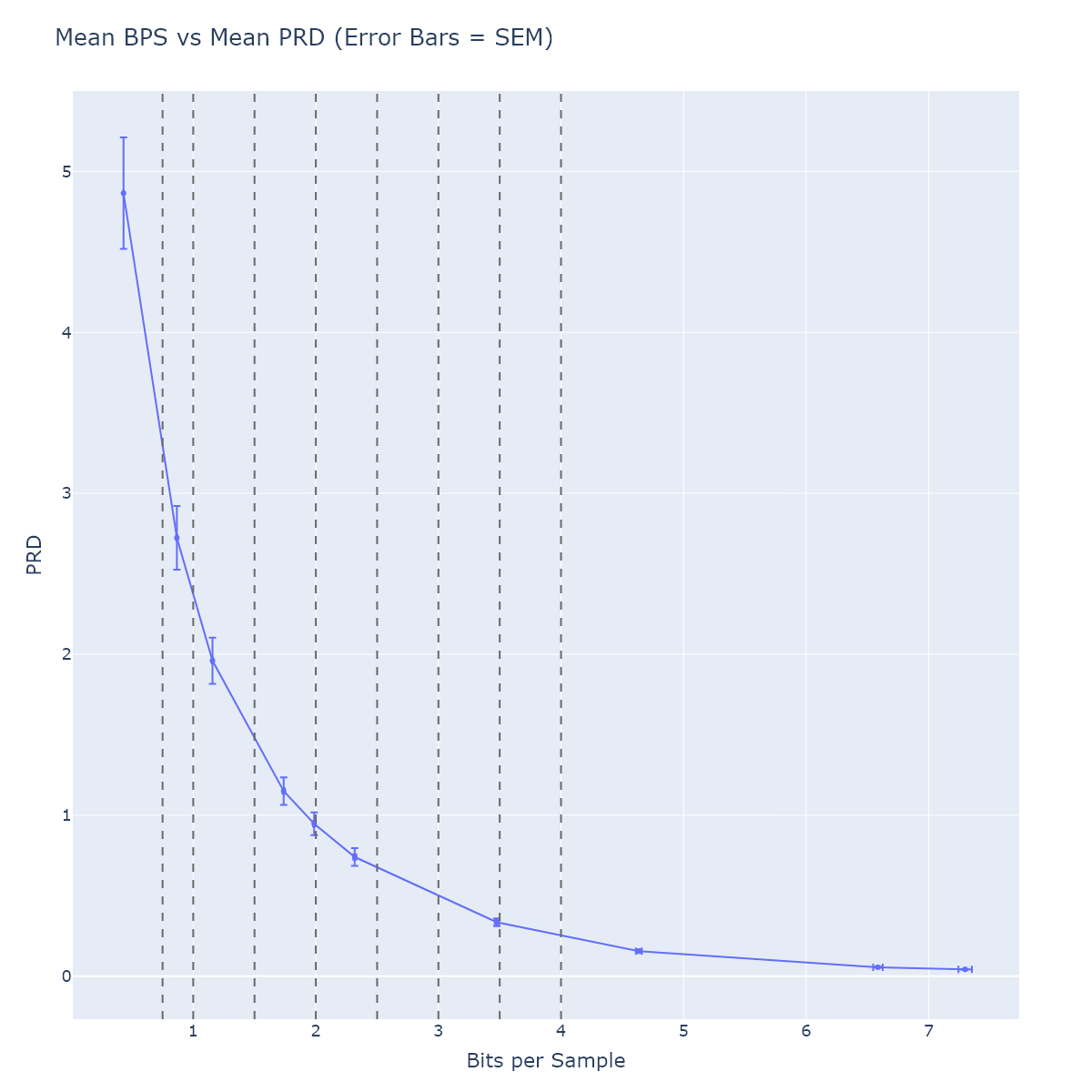


Figure 9 - Mean PRD values vs mean BPS per compression level. Error bars indicate standard error of the mean.

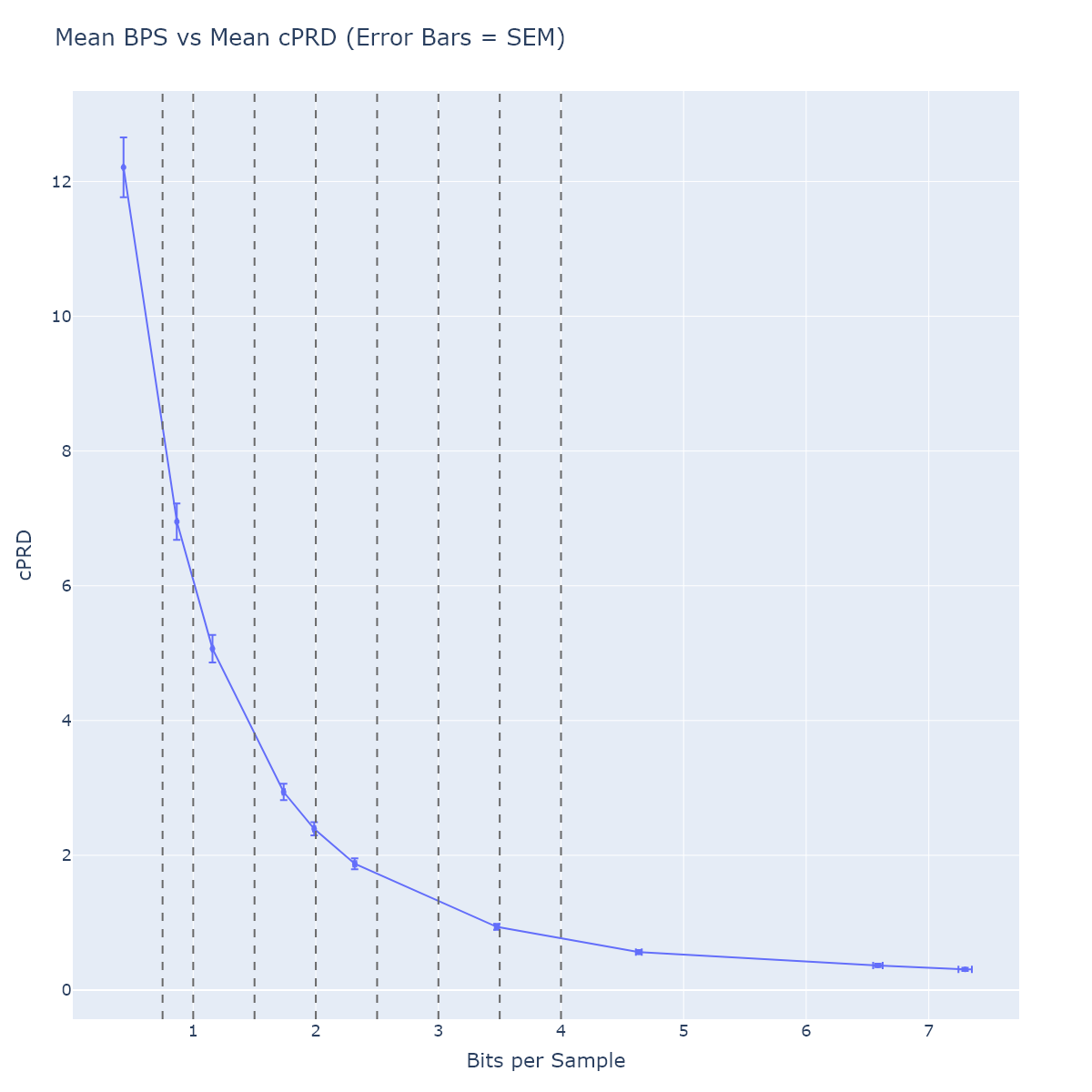


Figure 10 - Mean CPRD values vs mean BPS per compression level. Error bars indicate standard error of the mean.

A graph with a line

Description automatically generated

Figure 11 - Mean PSNR values vs mean BPS per compression level. Error bars indicate standard error of the mean.

A graph of a graph

Description automatically generated with medium confidence

Figure 12 - Example of the effect of frame duration on bitrate. Showing CPRD for the ECG dataset with frame durations of 10 seconds (red), 60 seconds (purple) and 300 seconds (green).

# Patent rights declarations(s)

Philips may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

# References

1. G.Sullivan and J. Pfaff, “Call for Proposals for a new ITU-T Recommendation H.BWC on the coding of biomedical waveform data” Q.6/SG16, *doc.*SG16-TD286-R1/PLEN, Rennes, April. 2024.
2. Manikandan, M. S. & Dandapat, S. Wavelet-based electrocardiogram signal compression methods and their performances: A prospective review. Biomedical Signal Processing and Control 14, 73–107 (2014)
3. A. Cohen, P. M. Poluta, and R. Scott-Millar, “Compression of ECG signals using vector quantization,” in Proc. IEEE-90 S. A. Symp. Communications and Signal Proc., COMSIG-90, 1990, pp. 45–54.
4. Zigel Y, Cohen A, Katz A. The weighted diagnostic distortion (WDD) measure for ECG signal compression. IEEE Trans Biomed Eng. 2000 Nov;47(11):1424-30. doi: 10.1109/TBME.2000.880093. PMID: 11077735.
5. <https://en.wikipedia.org/wiki/Pan%E2%80%93Tompkins_algorithm>
6. ISO/IEC 15444-1/ITU-T Rec. T.800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_