DOI
10.5281/zenodo.3908174

Publisher
CEUR-WS

Link to publication
http://ceur-ws.org/Vol-2626/paper9.pdf

Lossy Compression Methods for Performance-Restricted Wearable Devices

Lucie Klus[a], Elena Simona Lohan[a], Carlos Granell[b] and Jari Nurmi[a]

[a] *Tampere University, Electrical Engineering Unit, Tampere, Finland*
[b] *Universitat Jaume I, Institute of New Imaging Technologies, Castellon, Spain*

Abstract
With the increasing popularity, diversity, and utilization of wearable devices, the data transfer-and-storage efficiency becomes increasingly important. This paper evaluates a set of compression techniques regarding their utilization in crowdsourced wearable data. Transform-based Discrete Cosine Transform (DCT), interpolation-based Lightweight Temporal Compression (LTC) and dimensionality reduction-focused Symbolic Aggregate Approximation (SAX) were chosen as traditional methods. Additionally, an altered SAX (ASAX) is proposed by the authors and implemented to overcome some of the shortcomings of the traditional methods. As one of the most commonly measured entities in wearable devices, heart rate data were chosen to compare the performance and complexity of the selected compression methods. Main results suggest that best compression results are obtained with LTC, which is also the most complex of the studied methods. The best performance-complexity trade-off is achieved with SAX. Our proposed ASAX has the best dynamic properties among the evaluated methods.

Keywords
Compression, Discrete Cosine Transform (DCT), Lightweight Temporal Compression (LTC), Heart Rate, Symbolic Aggregation Approximation (SAX), Wearables

1. Introduction

During the last decade, wearable devices have become increasingly popular, with smart watches and wristbands dominating the market [1]. Wearables have found their utilization in a wide variety of fields, including medical research and patient monitoring, person tracking for enhanced security, and various applications in industrial halls. As the size of the devices is one determining factor in a wearable product, the battery requirements are typically very strict, leading to limited computational capabilities in the sense of hardware- or software-restricted processing power and data storage.

The amount of data produced by wearable devices increases continuously, especially in health monitoring applications, where physiological data such heart rate, ECG, breath rate, body temperature etc. are often monitored to prevent certain diseases, to track patients with long-lasting illnesses, or to monitor the changes in one's health. With the threat of more infectious diseases such as COVID-19 in the future, also location-related data and person tracking information are likely to more and more harnessed from wearables for the purpose of disease control and management. Long-term monitoring requires the storage of data for a long time in order to

ICL-GNSS 2020 WiP Proceedings, June 02–04, 2020, Tampere, Finland
EMAIL: lucie.klus@tuni.fi (L. Klus); elena-simona.lohan@tuni.fi (E.S. Lohan); carlos.granell@uji.es (C. Granell); jari.nurmi@tuni.fi (J. Nurmi)
ORCID: 0000-0003-0354-9421 (L. Klus); 0000-0003-1718-6924 (E.S. Lohan); 0000-0003-1004-9695 (C. Granell); 0000-0003-2169-4606 (J. Nurmi)

be able to detect the changes over the span of months or years, while the frequency of certain measurements is required to be high (e.g. heart-connected measurements). In order to detect heart attack, for example, heart measurements should be conducted within the intervals of a few seconds, as in these cases high frequency is the key, because heart rate changes are rapid and high. The demand for long-term monitoring comes not only from the medical side but also from the patient's side, as more and more patients acknowledge the advantages of prevention and (cardiac) rehabilitation [2].

Although the current wearable market for health applications offers a variety of heart trackers, most of them are not as precise as the heart monitors available in hospitals or are not suitable for everyday usage (requiring a belt attached to the chest). The precision of the measurements of wrist-worn devices is strongly dependent on the implemented sensors and it is affected by many additional factors, such as hand movements and how tightly the device is attached to the wrist.

In [3], the authors conducted a comparison of 17 heart rate trackers regarding their accuracy and precision. They found out that the accuracy of the trackers was between 79.8 % and 99.1 %, concluding that wearable devices are able to collect measurements with reasonable accuracy.

As the measurements themselves are reflecting the biometric data with a certain error, reducing the size of gathered data using approximation, thinning, or lossy compression is acceptable. Nevertheless, certain restrictions on the delay or error have to be applied, mostly based on the purpose of the device (medical monitoring vs. activity tracking, etc.).

The amount of data gathered increases with the number of sensors integrated in a wearable. Considering, for example, that a device conducts a measurement every 10 seconds, the resulting total amount of measurements can add up to 8640 measurements per day and about 3153600 measurements per year and per user. Each measurement requires to store at least the measured value and its timestamp. The most basic measurements may be stored in 4-byte formats, with timestamp varying between 7 to 13 bytes. In case of 4 bytes per measurement plus 7 bytes per timestamp, each entry for each sensor requires minimum of 11 bytes to be stored, resulting up to 34.7 megabytes of data per sensor per user per year.

The majority of the sensor data from wearables are time-series data, which are usually highly correlated in time. Therefore, representing such data by sampling between correlated samples or by other dimensionality-reduction methods may remove redundancy and reduce noise within the data. Based on the above considerations, there are several reasons to apply lossy compression on the data. In this paper, four most promising lossy compression techniques are evaluated based on their compression ratio capabilities, performance requirements and data distortion due to the compression. In order to select these four methods, an initial literature review has been performed by the authors and three of the presented methods were selected based on their good reported properties in the current literature. The fourth considered compression technique was developed by the authors.

2. Chosen Methods - Theory and properties

This section presents and describes the lossy compression methods utilized in this paper. The considered methods include Discrete Cosine Transform (DCT) with Run Length Encoding (RLE), Symbolic Aggregate Approximation (SAX), Altered Symbolic Aggregate Approximation (ASAX) proposed by the authors for the purposes of this paper and Lightweight Temporal Compression (LTC).

2.1. Discrete Cosine Transform with Run-Length Encoding (DCT-RLE)

Discrete Cosine Transform (DCT) is a transform-based compression widely utilized in media compression, including JPEG, MPEG, and various other formats. DCT has strong filtering properties, being able to reduce also the noise while reducing the size of the data. For these properties, DCT is utilized in many applications, including wireless positioning [4], where the authors used spectral analysis in the RSS-based indoor positioning scenario. The implemented DCT was used to compress the database size of RSS measurements with up to 80 % compression ratio, while achieving the performance of the conventional fingerprinting. DCT was also used to retrieve medical images in [5] and finds its utilization in medical images watermarking schemes [6][7].

As the core component of the DCT compression, a DCT transformation is used on the input data (metrics in the shape of $N \times M$). It results in the matrix of coefficients of the same shape ($N \times M$) as the input, referring to the cosine-shaped trends of different frequencies present within the data. Eq. (1) shows the formula for calculating the resulting coefficients of the transformation, where u and v are the coordinates in the coefficient matrix, M and N are the dimensions of input matrix (data block) and $f(x, y)$ is the input matrix value at x and y coordinates [8].

$$F(u,v) = \frac{2}{\sqrt{NM}} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) cos\left[\frac{(2x+1)u\pi}{2N}\right] cos\left[\frac{(2y+1)u\pi}{2M}\right] \quad \text{for } u = 0, ..., N-1 \text{ and } v = 0, ..., M-1 \tag{1}$$

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \\ 1 & \text{if } k > 0 \end{cases}$$

During the compression, the low-impact components, usually representing the higher frequencies in the data having close-to-zero values, are set to zero, based on the utilized algorithm. The compression ratio, as well as the resulting loss, can be varied based on the number of components forced to be zero. The data can be reconstructed using an inverse Discrete Cosine Transform (iDCT), which transforms the matrix of coefficients back to the original data.

From the computational point of view, the method is rather expensive [9]. In this paper we refer to combination of DCT with lossless compression in order to reduce the amount of stored data, as DCT itself does not decrease number of samples, but it rather transforms the data, resulting in many close-to-zero values, which may be set to zero. Such a result may be stored more efficiently by using lossless compression which reduces the redundancy in the data.

Run-Length Encoding (RLE) is such a lossless compression method. It stores repetitive values of input signal as a value and number of repetitions instead of storing every value separately. There are several ways of implementing the method. For the purposes of this paper, we apply the RLE per-sample. We stored the data as two same-length vectors, one with values of coefficients, the second one with their counts, as the output of the compression.

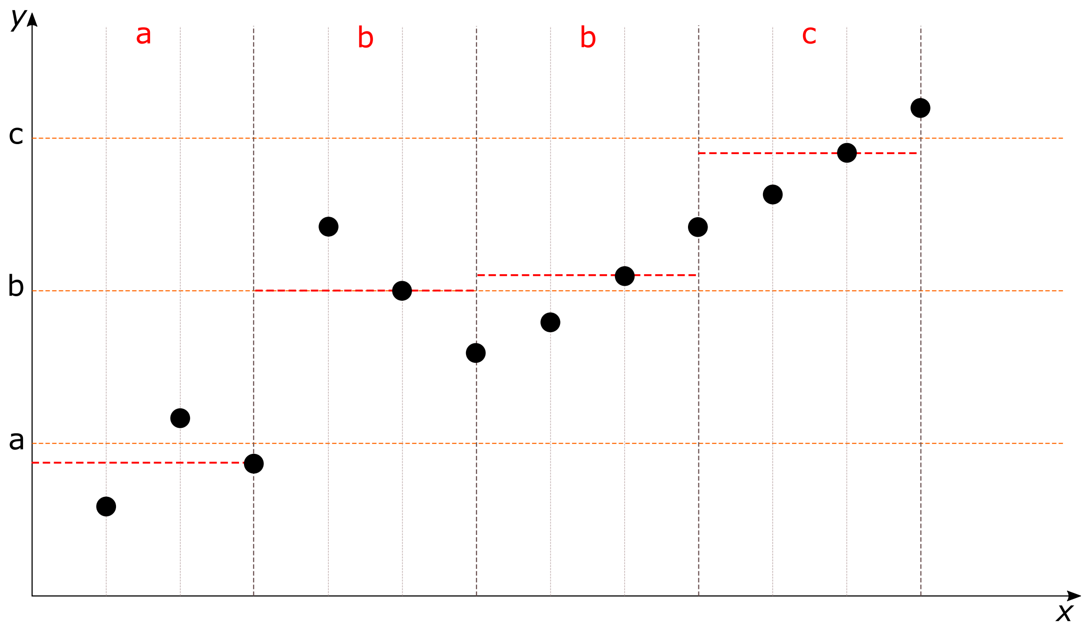

Figure 1: Illustration of SAX Implementation.

2.2. Symbolic Aggregate Approximation (SAX)

Symbolic Aggregate Approximation is a lossy compression method, composed of two basic steps. The first step is a Piece-wise Aggregate Approximation (PAA) compression, and the second one is a discretization of the y-axis and alphabet symbol assignment [10].

PAA compression is a traditional and non-complex type of compression (see Fig. 1), which takes a fixed range of samples (fixed amount of samples or fixed time interval), calculates their mean and returns it as the result for the corresponding interval (red dashed lines in the Figure). Next, SAX discretizes the y-axis into bins, e.g. (a, b, c) displayed as orange lines and assigns the PAA output (red lines) to the closest alphabet value (orange line) as an output (a, b, b, c).

The vertical distance between the bins is in most applications derived based on the distribution of the PAA data, as the normalized time series is assumed to have its distribution close to the Gaussian [10]. The resulting levels are then set so that each alphabet letter has the same probability of occurrence. In our implementation, the alphabet is based on the equidistant values of y-axis. The final result is stored as the output values of y-axis bins, time stamp of first sample and the length of equidistant sample range of x-axis.

The main advantage of SAX is the ability to compress data in both x and y axis, resulting in better compression capabilities compared to the transform-based methods such as DCT methods. In the literature, SAX is shown to be a lossy compression with wide data-mining utilization (e.g. clustering, classification or anomaly detection), while requiring less storage due to the limited alphabet size.

2.3. Altered Symbolic Aggregate Approximation (ASAX)

Altered Symbolic Aggregate Approximation is a compression method proposed by us and inspired by lossy SAX and lossless RLE. Based on SAX, it incorporates the discretized y-axis and from RLE the redundancy reduction due to the limited number of symbols. ASAX is, in

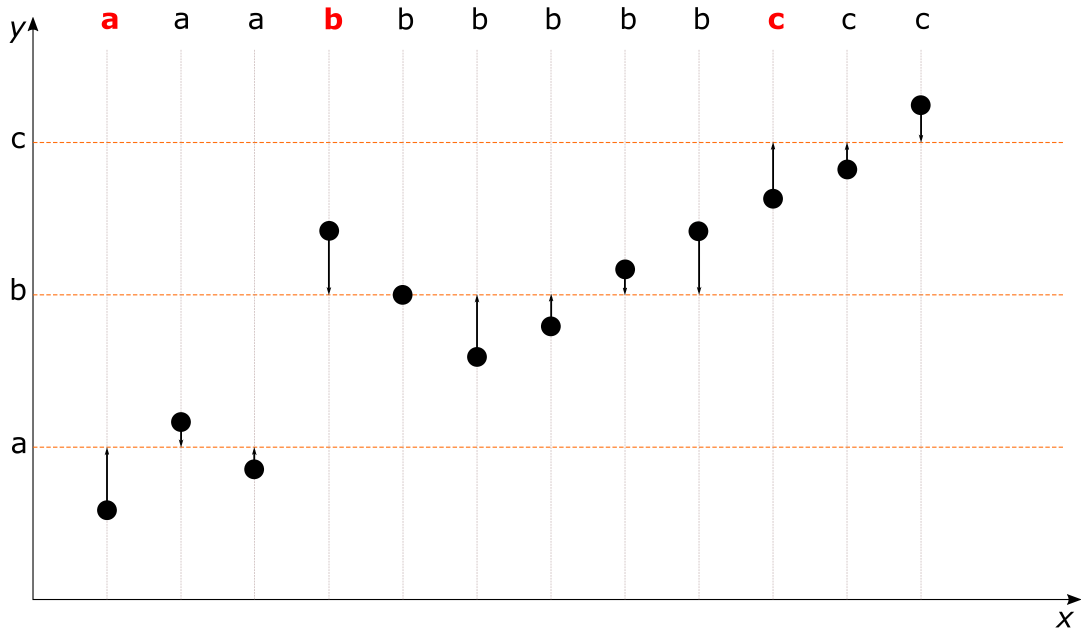

Figure 2: Illustration of the proposed ASAX implementation.

our opinion, also similar to Adaptive Piecewise Constant Approximation (APCA), proposed in [11], while limiting the alphabet size instead of averaging the sequences.

In comparison to SAX, ASAX method does not apply PAA, and it directly transforms the input values to the alphabet levels. Each sample is directly assigned to the discretized level closest to its original y-axis value, while x-axis value remains the same (see Fig. 2). The only parameter of this method is δ, representing the discretization step of the y-axis and determining the compression ratio. In the next step, all samples but the first one sharing the same alphabet symbol are excluded, until the change of symbol occurs. The output of the compression is the sequence of all remaining samples and their time stamp (x-axis value).

The authors proposed this method to maximize the ability to adapt to the dynamic changes in the data. The ASAX method, compared to the original SAX, allows for more dynamic system reactions in case a significant change occurs (e.g. heart attack), rather than waiting for the entire block of samples to be measured, averaged, assigned and only then sent. On the other hand, this method does not compress the x-axis equidistantly, causing the compression ratio to be highly dependent on the data, as well as the chosen parameter delta. Therefore, the compression ratio is greater for data with slower changes compared to SAX method.

2.4. Lightweight Temporal Compression (LTC)

Lightweight Temporal Compression is based on a piece-wise linear approximation. Its main advantages are high compression rate capabilities and fast processing speed, i.e., low demands on CPU and memory. Resembling RLE which compresses long strings of repeating symbols by a single symbol, the LTC compresses the data based on the linear trend that occurs in them [12].

The LTC algorithm is simple to implement and the basic principle is captured in Fig. 3, where the black dots represent the input samples in the time-series, Z_i represents the current

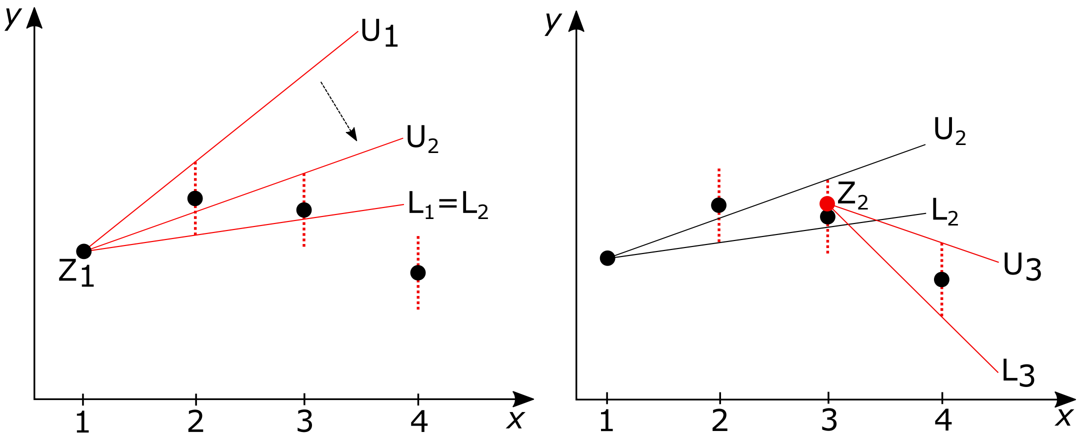

Figure 3: Illustration of LTC implementation.

output sample, U_i and L_i represent the upper and lower bound of the compression, respectively. It also has a single adjustable parameter called δ (red dashed lines), representing the tolerance of the method to the changes within data. Sample 1 (marked as Z_1) is the first symbol of input data. Connecting Z_1 with the extremes of the δ range of the following sample (coordinates [x_2, y_2+δ] and [x_2, y_2-δ]) creates the upper and lower bounds for the compression, called U_1 and L_1 respectively.

In case the δ-range (between [x$_3$, y_3+δ] and [x_3, y_3-δ]) of the following (third) sample intersects with the area in between U_1 and L_1 (see Fig. 3, left), a new set of bounds is created (U_2 and L_2). In case the upper point ([x_3, y_3+δ]) lies below U_1, U_2 is created by connecting Z_1 to the upper point. In case the lower point ([x_3, y_3-δ]) lies above the L_1, L_2 is created by connecting Z_1 to the lower point of δ-range.

In case the δ-range of the following sample lies outside of the area between two bounds (see Fig. 3, right), a new initial point (Z_2) is set. It is placed in the middle of the current bounds (U_2 and L_2) at the last valid δ-range line (see Eq. (2)). A new set of bounds (U_3 and L_3) are created by connecting Z_2 to the upper and lower point of the current sample's δ-range.

$$Z_2 = \left[x_3, \frac{U_2(x_3) + L_2(x_3)}{2} \right] \tag{2}$$

The compression is completed by repeating the process. The output of the compression is the sequence of all Z-point coordinates.

The algorithmic expression of how the method is implemented can be found in [12]. LTC was already evaluated to be feasible for wireless sensor networks. It allows to sample the input data at a high rate, it requires low data storage and it assists with noise suppression [13]. The method is extendable into multi-dimensional space [14]. One of possible alternatives to the method is its altered version called Refined Lightweight Temporal Compression (RLTC). Its compression ratio and energy requirements are improved in comparison to the traditional LTC, while its memory usage and latency overhead are bigger [15], which is why we do not utilize it in this paper.

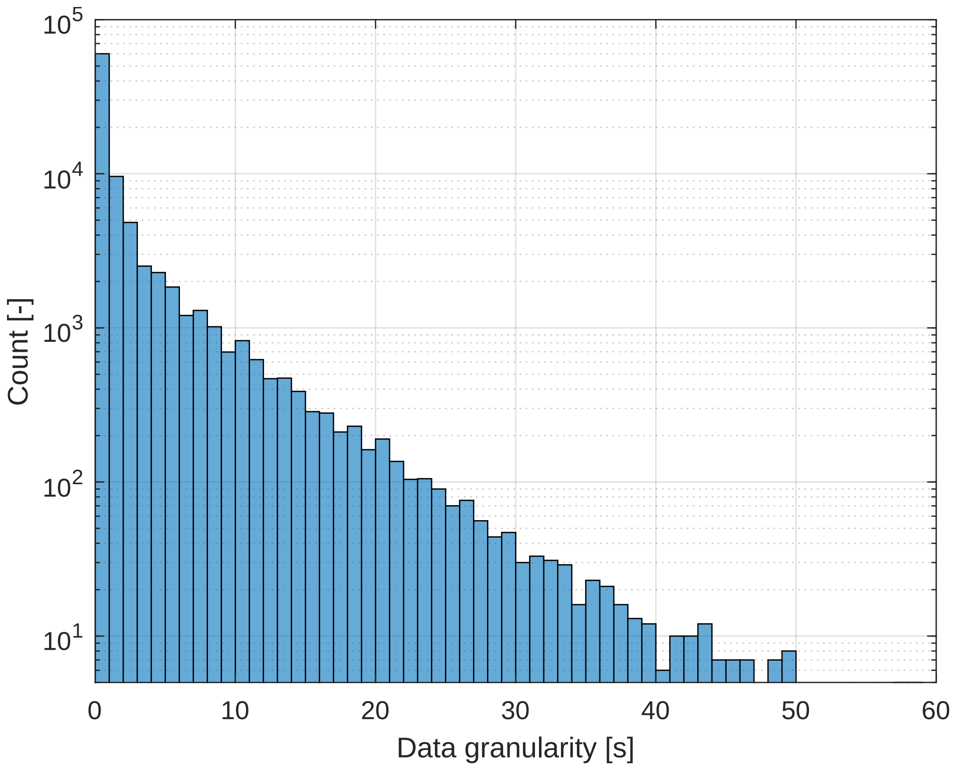

Figure 4: The granularity of heart rate measurements.

3. Dataset description for methods evaluation

The dataset titled *An Open Dataset for Human Activity Analysis using Smart Devices*, described in [16] was used for the evaluation of compression methods. The dataset was previously utilized in [17] and [18]. The whole available collection of three datasets consists of measurements by three wearable devices (phone, watch and glasses) of one subject, taken over 15 consecutive days (from morning to evening). Only the data measured by the smartwatch are considered, including only its timestamps and heart rate values (beats per minute). The LG Watch Urbane 2 type of smartwatch was used. The data consists of 91337 heat rate measurements and is further explained in [16]. The granularity of these measurements may be seen in Fig. 4. The dataset was chosen due to the high frequency of measurements, large number of entries and availability of the dataset. We remark that an initial comparison of various wearable datasets was done by the authors in [19].

The selected dataset was pre-processed by removing all zero values, not-a-number values, etc. (total of 9 values). Due to the high frequency of measurements, the heart rate values of two neighboring samples are not changing rapidly, resulting into slow changes and high correlation in the data as well as temporally linear segments in the course of the data.

4. Comparison of methods based on the selected wearable dataset

4.1. Performance metrics

To compare the compression performance, computational complexity and long-term storage suitability of the compression methods, we utilize three basic metrics. The first one of them, Root Mean Squared Error (RMSE), represents the degree of the distortion of the original data compared to the reconstructed data (see Eq. (3), where N stands for number of samples in each vector, s_n is n^{th} sample of original data vector and r_n is n^{th} sample of the reconstructed vector). To represent the compression distortion more universally, the resulting error was converted to % values by dividing it with y-axis range of the original data $max(s) - min(s)$ and multiplying by 100.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}[s_n - r_n]^2} \cdot \frac{100}{max(s) - min(s)} \tag{3}$$

Next, the Compression Ratio (CR) (see Eq. (4)) reflects the degree of compression of the method, represented by number of samples in equivalent data format.

$$CR = \frac{size(original\ data)}{size(compressed\ data)} \tag{4}$$

Finally, Compression Time (CT) is employed to evaluate the computational complexity of each method, represented by time required to compress the single data sample during the full dataset compression. Big-O notation is in this case non-explanatory since the compression of time-series data is preferably executed per-sample (or per-sample batch).

4.2. Performance comparison of the methods

In the following paragraphs, the methods are compared and evaluated based on the chosen performance metrics. The comparison was generated under the assumption of transmitting data approximately every 15 minutes, resulting in 100 samples per DCT-RLE block, while measuring the data every 8-10 seconds. Regarding the compression efficiency, the LTC gives the best results when comparing RMSE to the CR, as shown in Fig. 5, with 1.9 % RMSE at a compression ratio of 10. The SAX performs only slightly worse than LTC with 2.2 % RMSE at a CR equal to 10, followed by ASAX with 2.9 % RMSE at the same CR = 10. DCT-RLE falls significantly behind the other methods with 21.6 % RMSE at CR = 10.

The evaluation of computational complexity of the chosen methods is shown in Fig. 6, where SAX performed the fastest for all CRs higher than 3, followed by ASAX. LTC and DCT-RLE perform approximately 4 times slower than ASAX. Considering the above, LTC is the method that trades slightly better compression results for higher computational complexity. ASAX performs the fastest at the low compression values, overtaken by SAX at the higher compression ratios. Both SAX and ASAX perform with moderate compression error compared to the two remaining techniques, showing the promising trade-off between CT and RMSE. DCT-RLE shows the poorest performance among the studied methods with regard to RMSE as well as computational complexity.

DCT, from its definition, is suitable for data with periodic changes as it is based on frequency-domain transformation. In case the measurements are not taken in time spans of seconds,

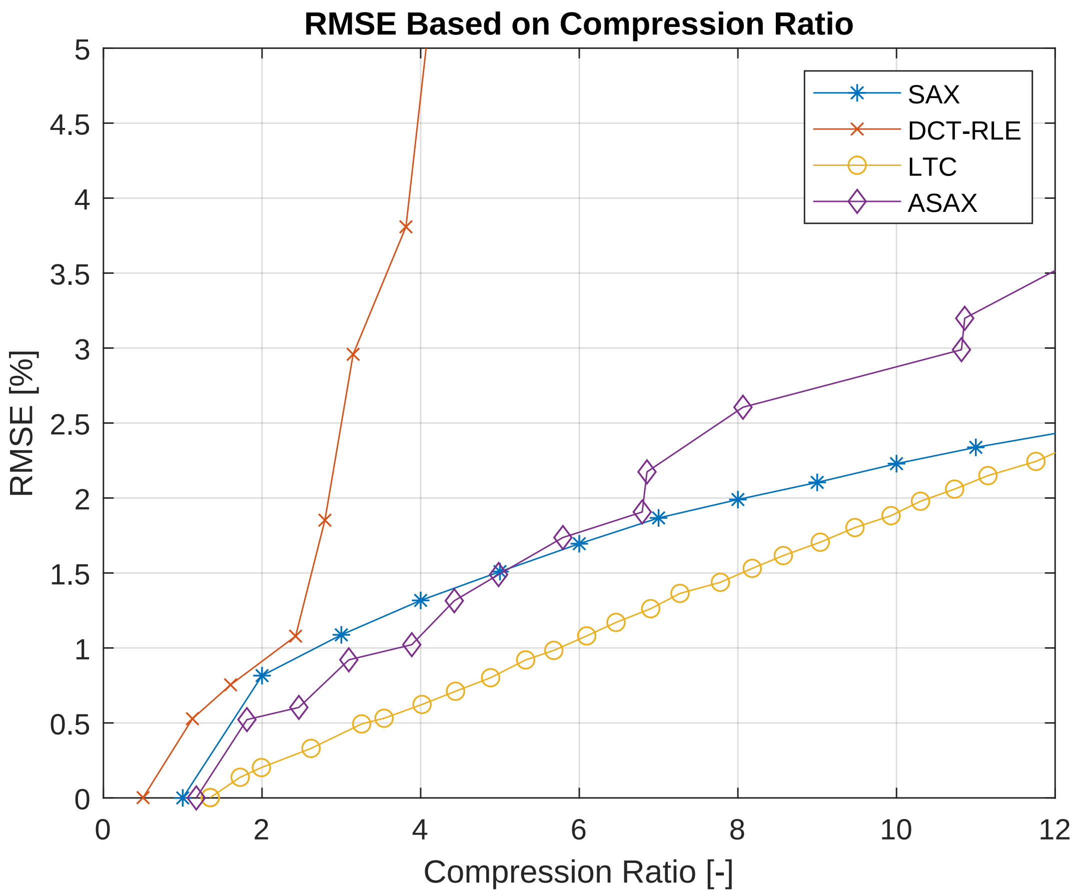

Figure 5: Comparison of RMSE to Compression Ratio of Different Methods

but rather minutes, the changes between neighboring samples start to vary too unexpectedly (e.g., the patient sleeps during one measurement and chases the bus during the next one), which would result in high dynamic error. Therefore, the precision of the decompression gets lowered, causing possible misinterpretation of the data. This case can be considered an advantage in some use cases of DCT, where it is able to filter out high frequencies (e.g. noise in measurements).

From the definition of the methods, in case of sudden high-frequency changes, SAX, similarly to the DCT, trims the extremes by averaging over the given set of samples (PAA part of the algorithm). ASAX however reacts instantaneously to the peak, immediately reporting the new alphabet value. LTC reacts to the sudden changes, but with certain delay as it always reports the last Z-value instead of reacting to the current sample. In case the wearable health application run on the sensor device is aiming to detect sudden changes (events such as cardiac arrest), the most suitable method of the above-studied ones is ASAX, since it is the one able to follow fast the rapid changes.

4.3. Transmission Frequency and Power Consumption

DCT-RLE requires the whole block of data to be compressed at once and therefore it is limited in terms of block size (or step). As the high frequency components are usually weaker,

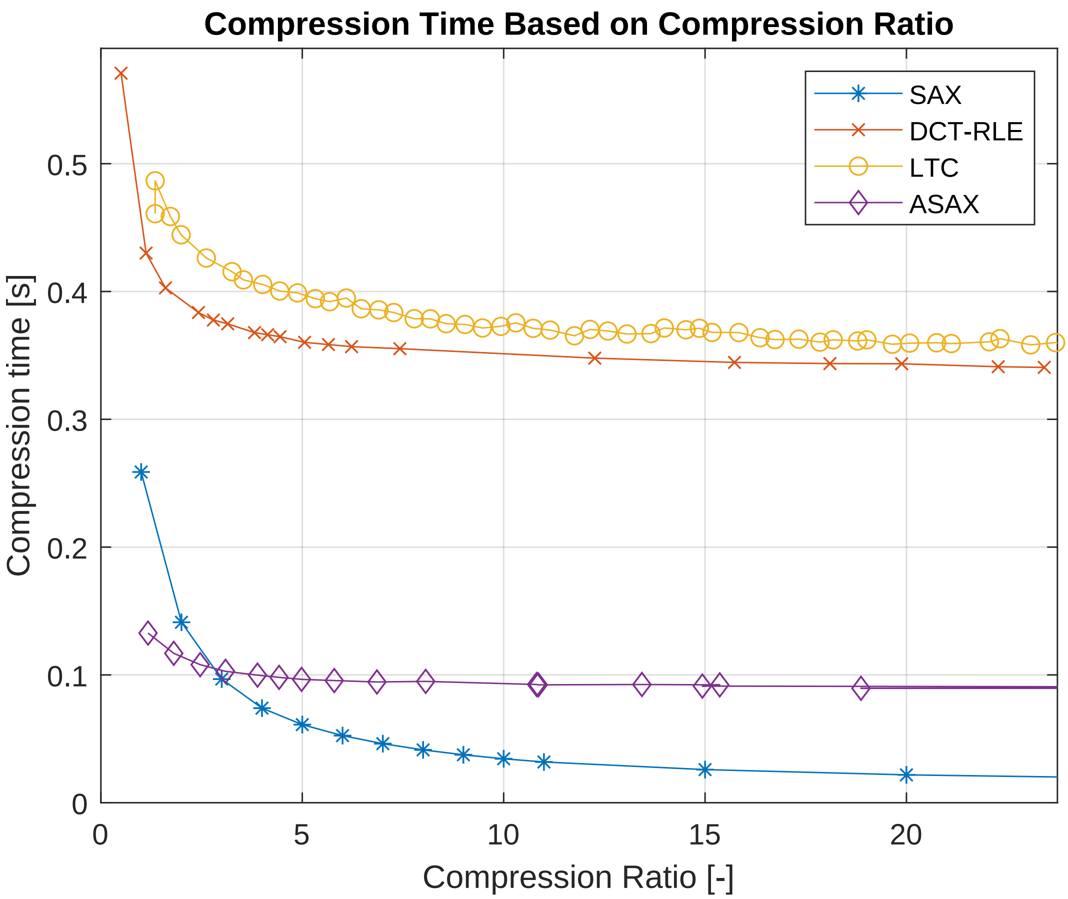

Figure 6: Comparison of Computation Time to Compression Ratio of Different Methods

DCT-RLE is efficient if applied to larger blocks of data, as shown below. Furthermore, the combination of DCT with RLE further improves the efficiency of large block size approach, as it compresses more zero-values into only one symbol. Fig. 7 visualizes the performance of DCT-RLE, based on the varying block size. It is clearly visible that the small block size limits the method, while compressing blocks of 5000 samples result in 3 % RMSE while the CR is 40, making the method suitable for long-term data storage applications.

Although the DCT-RLE utilization of bigger block sizes results in higher CR while having smaller RMSE, DCT-RLE is not suitable for applications with high frequency of data transmission. For the measurements taken every 8 to 10 seconds, approximately 100 samples are produced every 15 minutes corresponding to the DCT block size equal to 100. This adds up to 9600 measurements produced per each day. Therefore, DCT may be applied in wearable devices that transmit the measurement once per day, but the utilization in low latency systems is not suitable. In case of low latency systems, ASAX is the most suitable presented method for wearables, as it reports the changes immediately, as explained above.

To evaluate the dependency of various DCT block sizes on CT, first 20000 samples from the evaluated dataset were utilized. Fig. 8 shows, that with the higher block size, the efficiency also increases with regard to the computational complexity, when evaluating the CT based on the CR. The CT of block size of 50 in comparison to the CT of block size 5000 is almost

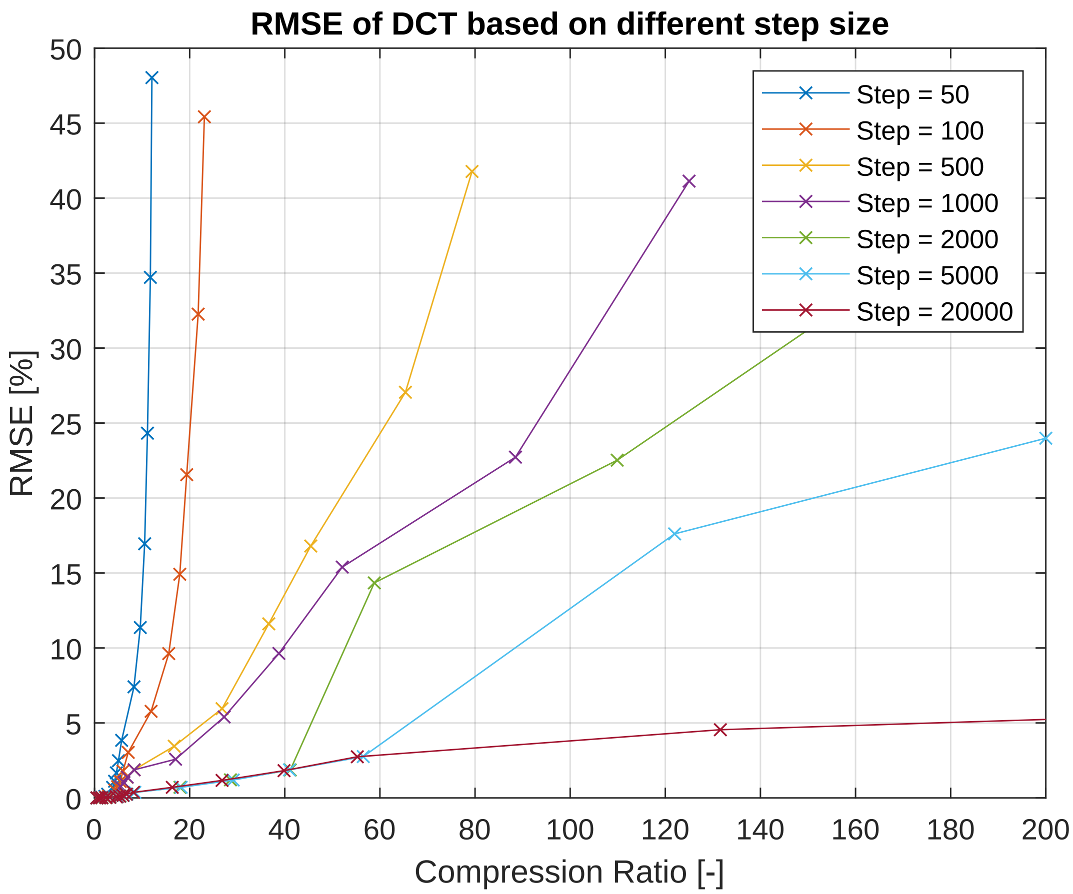

Figure 7: Comparison of RMSE to Compression Ratio of DCT

ten times higher. Lines referring to smaller step sizes are limited in CR, as high CRs are impossible to obtain with limited block size. This result supports the claim, that DCT-RLE is highly efficient in case the latency of the system is not an issue. Unlike DCT-RLE, the other three studied algorithms, ASAX, SAX, and LTC are not highly dependent on the block size, as they are compressing in more dynamic fashion.

An additional point to consider when dealing with wearable devices and their power efficiency is the power consumption of each connection and transmission of the data to a gateway node. Frequent reporting leads to larger power consumption overhead due to communication initiation cost.

4.4. Long-term Storage

On one hand, for the immediate diagnosis (e.g. in case of heart attack), single value measurements taken at high frequency are important. These are also essential during activity tracking, especially for high-performance exercise evaluation. On the other hand, the data stored for a long time (e.g. years) may not have such information gain compared to the amount of data stored in the short-term evaluation. For the purposes of long-term patient tracking, approximate heart rate measures may be sufficient.

This way, the input alphabet size for the compression could be decreased and therefore the

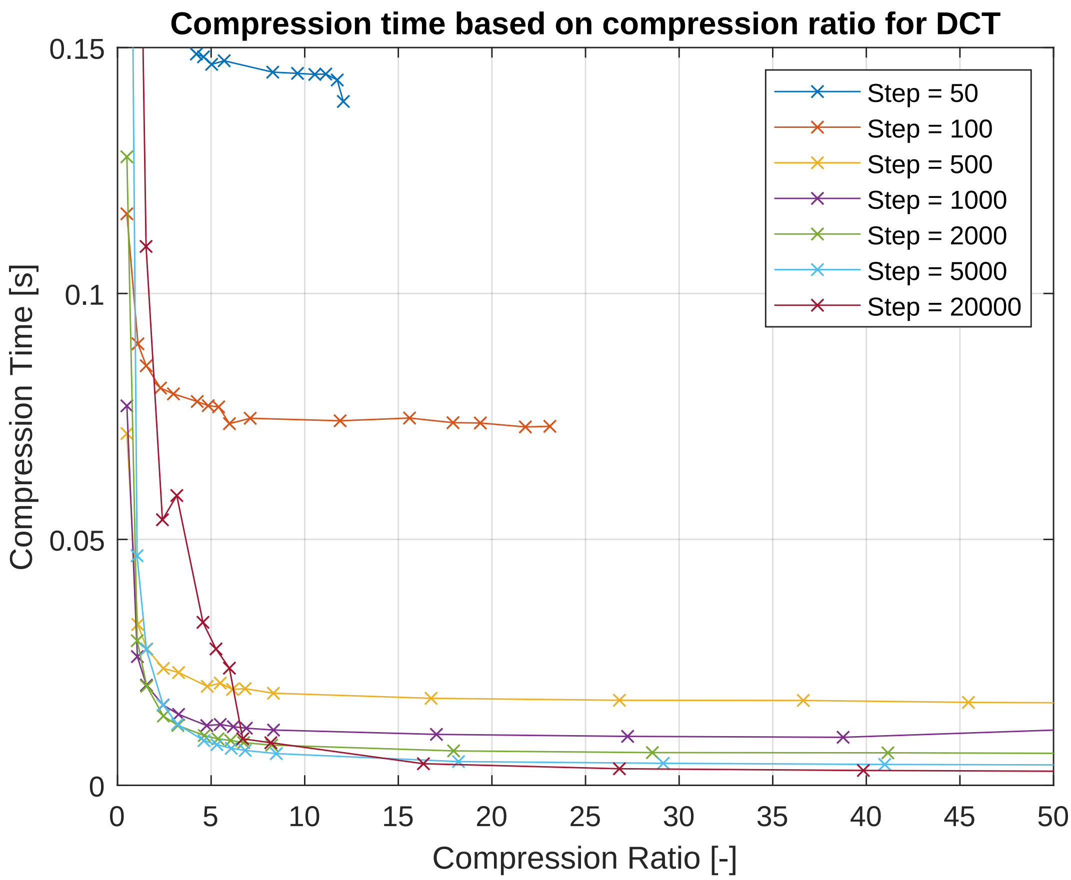

Figure 8: Comparison of Computation Time to Compression Ratio of DCT

data size would be significantly reduced. For this purpose, a two-stage approach should be utilized. The lossless compression, such as RLE, Huffman Coding (HC) or Lempel-Ziv-Welch (LZW), should follow the lossy one. Therefore, another possible application of DCT-RLE with big block size based on our studies is in the long-term storage. SAX can also be an efficient technique for long-term storage, as the fixed time-step and limited alphabet size allow for highly efficient bit-level compression in both dimensions. LTC and ASAX are rather lacking in terms of data storage, as both methods and their compression efficiency are highly dependent on the dynamics of the data they are applied to.

4.5. Results summary

The summary of the suitability of different methods for the chosen applications can be found in Table 1. DCT-RLE is considered twice there, once with ”Small Step”, referring to block sizes smaller than 100 for the applied dataset and ”Big Step”, referring to the block sizes of 2000 and more. The table gives a recommendation regarding the suitability of each method for different use-cases considered in this paper. The suitability ranges from ”Suitable” to ”Not Suitable”, with ”Neutral” referring to the cases, where the better option is available, yet the performance is sufficient.

Table 1
The results summary

	DCT-RLE Small Step	DCT-RLE Big Step	SAX	LTC	ASAX
Compression performance	✗	✓	-	✓	-
Computation complexity	✗	✓	✓	✗	-
Long-term storage	✗	✓	✓	-	-
Low-latency systems	-	✗	-	-	✓

✓Suitable ✗Not suitable -Neutral

5. Conclusions

The power consumption efficiency, as one of the main challenges of wearable-based devices, can be improved by applying lossy compression techniques to the wearable sensor-based data. This paper evaluates four different compression methods with regard to their applicability in such devices on heart rate data, as most commonly measured biometric data from wearables. The analysis shows that LTC performs the best with regard to the compression error, at the cost of a high computational complexity. SAX performs the fastest, when compression ratio is above 3, surpassed by ASAX at CR below 3. Both methods perform moderately regarding compression distortion metric (RMSE per compression ratio of the method). The proposed ASAX compression technique offers the best tradeoff between different performance metrics if deployed in a system requiring short latency. This is because ASAX reacts to the significant changes within the data without delay, unlike the other studied methods where the reaction times are higher. The performance of DCT-RLE is strongly dependent on the block size, determining the length of the vector fed to the algorithm. With shorter block size, e.g., below 1000 samples, DCT-RLE performs significantly worse than the remaining techniques. On the other hand, with block size 10000 or higher, the DCT-RLE is able to achieve very high compression ratios with low RMSE, making the DCT-RLE technique suitable for systems tolerating very high latency and for long-term storage of data. The ability of SAX to compress data in two dimensions makes it suitable for long-term compression as well.

Acknowledgments

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Sklodowska Curie grant agreement No. 813278 (A-WEAR, http://www.a-wear.eu/). Also, the authors would like to thank to Roman Klus for his help and valuable impacts during the preparation of this paper.

References

[1] E. Park, User acceptance of smart wearable devices: An expectation-confirmation model approach, Telematics and Informatics 47 (2020) 101318.
[2] K. Lauber, G. Wort, F. Gillison, D. Thompson, D. Augustine, O. Peacock, Patient views

of the role of wearable technology to enhance cardiac rehabilitation, in: United Kingdom Society for Behavioural Medicine annual conference, 2020.

[3] F. El-Amrawy, M. I. Nounou, Are currently available wearable devices for activity tracking and heart rate monitoring accurate, precise, and medically beneficial?, Healthcare informatics research 21 (2015) 315–320.

[4] J. Talvitie, M. Renfors, E. S. Lohan, Novel indoor positioning mechanism via spectral compression, IEEE Communications Letters 20 (2015) 352–355.

[5] C. Zhang, J. Li, S. Wang, Z. Wang, An encrypted medical image retrieval algorithm based on dwt-dct frequency domain, in: 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications, 2017.

[6] R. Thanki, S. Borra, V. Dwivedi, K. Borisagar, An efficient medical image watermarking scheme based on fdcut–dct, Engineering science and technology, an international journal 20 (2017) 1366–1379.

[7] A. Mehto, N. Mehra, Adaptive lossless medical image watermarking algorithm based on dct & dwt, Procedia Computer Science 78 (2016) 88–94.

[8] A. Raid, W. Khedr, M. A. El-Dosuky, W. Ahmed, Jpeg image compression using discrete cosine transform-a survey, arXiv preprint arXiv:1405.6147 (2014).

[9] D. Del Testa, M. Rossi, Lightweight lossy compression of biometric patterns via denoising autoencoders, IEEE Signal Processing Letters 22 (2015) 2304–2308.

[10] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing sax: a novel symbolic representation of time series, Data Mining and knowledge discovery 15 (2007) 107–144.

[11] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Locally adaptive dimensionality reduction for indexing large time series databases, in: Proc. of the 2001 ACM SIGMOD international conference on Management of data, 2001.

[12] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, D. Estrin, Lightweight temporal compression of microclimate datasets (2004).

[13] R. Sharma, A data compression application for wireless sensor networks using ltc algorithm, in: IEEE International Conference on Electro/Information Technology, 2015.

[14] B. Li, O. Sarbishei, H. Nourani, T. Glatard, A multi-dimensional extension of the lightweight temporal compression method, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 2918–2923.

[15] O. Sarbishei, Refined lightweight temporal compression for energy-efficient sensor data streaming, in: 2019 IEEE 5th World Forum on Internet of Things, IEEE, 2019.

[16] S. Faye, N. Louveton, S. Jafarnejad, R. Kryvchenko, T. Engel, An open dataset for human activity analysis using smart devices (2017).

[17] A. Das Antar, M. Ahmed, M. A. R. Ahad, Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review, 2019. doi:10.1109/ICIEV.2019.8858508.

[18] H. Alrehamy, C. Walker, (personal data lake) semlinker: automating big data integration for casual users, Journal of Big Data 5 (2018). doi:10.1186/s40537-018-0123-x.

[19] L. Klus, E. S. Lohan, C. Granell, J. Nurmi, Crowdsourcing solutions for data gathering from wearables, in Proc. of XXXV Finnish URSI Convention on Radio Science, 2019.