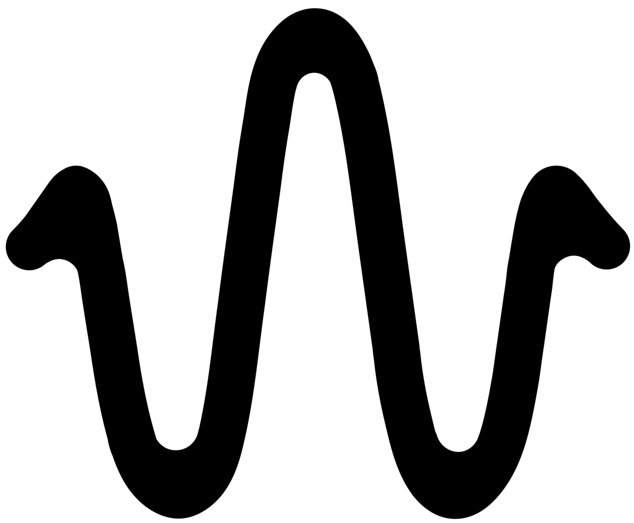
ECG Signal Compression: Road to the Proper Transform

Francesco Rosnati

August 7, 2024



Abstract

The aim of this study is to explore and compare various mathematical transforms, including the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT), to determine their suitability for compressing Electrocardiogram (ECG) signals. Each transform has unique properties that affect its performance in signal compression. By analyzing their computational complexity, energy compaction efficiency, and the nature of compression artifacts, this study seeks to identify the most effective transform for ECG signal compression. Special emphasis is placed on the DWT due to its inherent ability to handle non-stationary signals with transient features, which are characteristic of ECG signals.

Contents

1 Fourier Transform

The Fourier Transform (FT) is a mathematical transformation used to analyze the frequency content of continuous signals. It transforms a continuous time-domain function $f(t)$ into a continuous frequency-domain function $\hat{f}(\omega)$.

The FT is defined as:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\, dt$$

Here:

- $f(t)$ is the original time-domain signal.
- $\hat{f}(\omega)$ is the Fourier transform of $f(t)$, representing the signal in the frequency domain.
- ω is the angular frequency (in radians per second).
- $e^{-i\omega t}$ is the complex exponential basis function.

1.1 Inverse Fourier Transform

The inverse Fourier Transform (iFT) transforms the frequency-domain function $\hat{f}(\omega)$ back into the time-domain function $f(t)$.

The iFT is defined as:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega t}\, d\omega$$

Here:

- $\hat{f}(\omega)$ is the frequency-domain representation of the signal.
- $f(t)$ is the original time-domain signal.
- $e^{i\omega t}$ is the inverse complex exponential basis function.
- The factor $\frac{1}{2\pi}$ ensures that the transformation is normalized.

1.2 Interpretation and Properties

- **Frequency Content**: The Fourier transform $\hat{f}(\omega)$ provides the amplitude and phase of each frequency component present in the original signal $f(t)$.
- **Linearity**: The FT is a linear operator. If a and b are scalars, and $f(t)$ and $g(t)$ are signals, then:

  $$F\{af(t) + bg(t)\} = aF\{f(t)\} + bF\{g(t)\}$$

  where F denotes the Fourier transform.

2 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) transforms a sequence of complex numbers $\mathbf{f} = \{f_0, f_1, \ldots, f_{N-1}\}$ into another sequence of complex numbers $\hat{\mathbf{f}} = \{\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{N-1}\}$, representing the frequency domain. The DFT is defined as:

$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-i\frac{2\pi}{N}jk} \quad \text{for} \quad k = 0, 1, \ldots, N-1$$

Here:

- $\hat{f}_k$ are the Fourier coefficients.
- f_j are the input data points.
- $e^{-i\frac{2\pi}{N}jk}$ is the complex exponential basis function.

2.1 Inverse Discrete Fourier Transform (iDFT)

The inverse DFT transforms the sequence $\hat{\mathbf{f}}$ back to the original sequence $\mathbf{f}$. The iDFT is defined as:

$$f_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}_k e^{i\frac{2\pi}{N}jk} \quad \text{for} \quad j = 0, 1, \ldots, N-1$$

Here:

- f_j are the reconstructed data points.
- $\hat{f}_k$ are the Fourier coefficients.
- $e^{i\frac{2\pi}{N}jk}$ is the inverse complex exponential basis function.

2.2 Matrix Representation

Both DFT and iDFT can be represented as matrix operations. The DFT matrix $\mathbf{F}$ of size $N \times N$ is defined as:

$$\mathbf{F}_{jk} = e^{-i\frac{2\pi}{N}jk}$$

Explicitly, the DFT matrix $\mathbf{F}$ can be written as:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-i\frac{2\pi}{N}1} & e^{-i\frac{2\pi}{N}2} & \cdots & e^{-i\frac{2\pi}{N}(N-1)} \\ 1 & e^{-i\frac{2\pi}{N}2} & e^{-i\frac{2\pi}{N}4} & \cdots & e^{-i\frac{2\pi}{N}2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-i\frac{2\pi}{N}(N-1)} & e^{-i\frac{2\pi}{N}2(N-1)} & \cdots & e^{-i\frac{2\pi}{N}(N-1)^2} \end{bmatrix}$$

The DFT operation can then be written in matrix form as:

$$\hat{\mathbf{f}} = \mathbf{F}\mathbf{f}$$

where:

$$\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}, \quad \hat{\mathbf{f}} = \begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{N-1} \end{bmatrix}$$

For the iDFT, the inverse DFT matrix $\mathbf{F}^*$ (complex conjugate transpose of $\mathbf{F}$) is defined as:

$$\mathbf{F}^*_{jk} = e^{i\frac{2\pi}{N}jk}$$

Explicitly, the inverse DFT matrix $\mathbf{F}^*$ can be written as:

$$\mathbf{F}^* = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\frac{2\pi}{N}1} & e^{i\frac{2\pi}{N}2} & \cdots & e^{i\frac{2\pi}{N}(N-1)} \\ 1 & e^{i\frac{2\pi}{N}2} & e^{i\frac{2\pi}{N}4} & \cdots & e^{i\frac{2\pi}{N}2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i\frac{2\pi}{N}(N-1)} & e^{i\frac{2\pi}{N}2(N-1)} & \cdots & e^{i\frac{2\pi}{N}(N-1)^2} \end{bmatrix}$$

The iDFT operation can then be written as:

$$\mathbf{f} = \frac{1}{N}\mathbf{F}^*\hat{\mathbf{f}}$$

2.3 Linear Operator

The DFT is a linear operator, meaning that:

$$\text{DFT}(af + bg) = a \cdot \text{DFT}(f) + b \cdot \text{DFT}(g)$$

where a and b are scalars, and f and g are signals.

2.4 Complexity of DFT

The direct computation of the DFT has a time complexity of $O(N^2)$. However, the Fast Fourier Transform (FFT) algorithm reduces this complexity to $O(N \log N)$ by exploiting symmetries and periodicities in the DFT computation.

The Cooley-Tukey algorithm, the most commonly used FFT, recursively divides a DFT of size N into smaller DFTs, achieving the reduced complexity. The recurrence relation for the time complexity $T(N)$ of the FFT is:

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

which solves to:

$$T(N) = O(N \log N)$$

2.5 Mapping Continuous to Discrete Parameters

In the context of DFT, continuous parameters are mapped to their discrete counterparts as follows:

- **Time**:
  - Continuous: t
  - Discrete: j (indices of sampled points)
- **Frequency**:
  - Continuous: ν (frequency in Hz) or $\omega = 2\pi\nu$ (angular frequency)
  - Discrete: k (frequency index)
- **Functions**:
  - Continuous: $f(t)$ (time-domain function)
  - Discrete: $f[j]$ (sampled data sequence)

2.6 Dimensionality of k: Why N Frequencies with N Samples

The dimensionality of k in the DFT is the same as the number of points N. This means that the DFT produces N frequency components from N time-domain samples. Each k corresponds to a specific frequency in the frequency domain.

Connection to Frequency

- **Frequency Index** k: The index k in the Discrete Fourier Transform (DFT) corresponds to the discrete frequency components. The DFT transforms a time-domain signal into its frequency components, with k representing these discrete frequency bins.
- **Frequency Resolution**: The frequency resolution of the DFT is determined by the sampling rate f_s and the number of points N. The k-th frequency component corresponds to the frequency $\frac{kf_s}{N}$, where f_s is the sampling frequency.
- **Nyquist Frequency**: The highest frequency that can be uniquely represented without aliasing is the Nyquist frequency, $\frac{f_s}{2}$. This corresponds to $k = \frac{N}{2}$ when N is even.
- **Symmetry in Real-Valued Signals**: For real-valued signals, the DFT produces N complex numbers, but only the first $\frac{N}{2} + 1$ components (including the zero frequency and Nyquist frequency) are unique. The remaining components are redundant due to the symmetry (complex conjugate) property of the DFT of real-valued signals.

Several reasons contribute to the dimensionality of k being equal to N:

- **Nyquist-Shannon Sampling Theorem**: Ensures a signal can be fully represented by its samples if sampled at a rate at least twice the maximum frequency.
- **Periodic Nature of DFT**: Assumes the discrete sequence is periodic with period N, leading to N unique frequency components.
- **Orthogonality of Basis Functions**: Uses N orthogonal complex exponential functions as basis functions, each corresponding to a unique frequency component.
- **Signal Reconstruction**: Requires N frequency components to fully reconstruct the original sequence, ensuring no loss of information.

3 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a transform similar to the Discrete Fourier Transform (DFT) but uses only real numbers and cosines. It is widely used in image and video compression (e.g., JPEG, MPEG) due to its properties that are particularly suitable for these applications.

3.1 Overview of the Discrete Cosine Transform (DCT)

The DCT represents a signal as a sum of cosine functions oscillating at different frequencies. It transforms a sequence of real numbers into a sequence of coefficients representing the signal in the frequency domain.

3.2 Types of DCT

There are several types of DCT, but the most commonly used are DCT-I, DCT-II, and DCT-III. The most frequently used variant in practical applications is DCT-II, often referred to simply as "the DCT."

DCT-II (The Most Common DCT)

For a sequence of N real numbers $x[n]$, where $n = 0, 1, \ldots, N-1$, the DCT-II is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \quad \text{for} \quad k = 0, 1, \ldots, N-1$$

Inverse DCT-II

The inverse DCT-II (often referred to as IDCT) is defined as:

$$x[n] = \frac{1}{N}\left(\frac{X[0]}{2} + \sum_{k=1}^{N-1} X[k] \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]\right) \quad \text{for} \quad n = 0, 1, \ldots, N-1$$

DCT-I

The DCT-I is defined for a sequence $x[n]$ of length N as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi}{N-1} nk\right) \quad \text{for} \quad k = 0, 1, \ldots, N-1$$

DCT-I is defined only for sequences of length $N \geq 2$ and is less commonly used due to boundary conditions.

DCT-III

The DCT-III, often referred to as the inverse DCT of DCT-II, is defined as:

$$x[n] = \frac{1}{2}X[0] + \sum_{k=1}^{N-1} X[k] \cos\left(\frac{\pi}{N}k\left(n + \frac{1}{2}\right)\right) \quad \text{for} \quad n = 0, 1, \ldots, N-1$$

3.3 Comparison Between DCT and DFT

- **Basis Functions**:
  - **DFT**: Uses both sines and cosines (complex exponentials) as basis functions, leading to complex coefficients.
  - **DCT**: Uses only cosines as basis functions, leading to real coefficients.
- **Symmetry**:

  - **DFT**: Typically applied to periodic signals.
  - **DCT**: Often applied to even-symmetric signals (or the signal is made even-symmetric by mirroring).

- **Energy Compaction**:
  - **DCT**: Generally more efficient at energy compaction for real-valued signals, meaning it can represent most of the signal's energy in a few coefficients.

- **Applications**:
  - **DFT**: Used in spectral analysis, filtering, and general signal processing.
  - **DCT**: Widely used in image and video compression (e.g., JPEG, MPEG) due to its efficient energy compaction.

3.4 Properties of the DCT

- **Orthogonality**: The cosine basis functions used in DCT are orthogonal.
- **Real-Valued Output**: For real-valued input signals, the DCT output is also real-valued.
- **Energy Compaction**: The DCT tends to concentrate the energy of the signal in a few low-frequency components, making it efficient for compression.

3.5 Complexity of DCT

Direct Computation

The direct computation of DCT for a sequence of length N involves N multiplications and $N-1$ additions for each of the N frequency components, resulting in a total complexity of:

$$O(N^2)$$

Fast Algorithms for DCT

Fast algorithms, similar to the Fast Fourier Transform (FFT), reduce the computational complexity of the DCT to:

$$O(N \log N)$$

These algorithms exploit symmetry properties and use divide-and-conquer approaches to achieve significant computational savings.

3.6 Energy Compaction and Low-Frequency Components in DCT

The Discrete Cosine Transform (DCT) has a key property known as energy compaction, where most of the signal's energy is concentrated in a few low-frequency components. This property is essential for efficient compression, as it allows significant data reduction while preserving the essential features of the original signal.

In the DCT, the index k represents the frequency component. Low values of k correspond to low-frequency components, which represent slow variations in the signal, while high values of k correspond to high-frequency components, representing rapid variations.

Frequency Interpretation

- $k = 0$: The basis function is a constant, representing the average value of the signal.
- Low k: Represent slow variations, such as $\cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\cdot 1\right)$.
- High k: Represent rapid variations, such as $\cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\cdot (N-1)\right)$.

Energy Compaction

The DCT's ability to concentrate energy in low-frequency components means that for many natural signals, including images and audio, most of the significant information can be captured with only a few coefficients. This makes the DCT highly efficient for compression purposes, as the majority of high-frequency coefficients (which represent fine details and noise) can be quantized more coarsely or discarded without significantly affecting the perceived quality of the signal.

3.7 JPEG Compression Example

In JPEG compression, the DCT is used to transform spatial domain data into frequency domain data. The steps involved are:

DCT Computation The image is divided into 8×8 blocks, and the DCT is computed for each block.

Quantization Step Each DCT coefficient is divided by a corresponding value in a quantization matrix and then rounded. This step effectively reduces the precision of less important coefficients, which often corresponds to higher frequencies. A predefined quantization matrix specifies how much each DCT coefficient should be divided by. Each DCT coefficient $X[k]$ is divided by the corresponding entry in the quantization matrix and then rounded to the nearest integer.

$$\text{Quantized\_Coeff}[j,k] = \text{round}\left(\frac{X[j,k]}{Q[j,k]}\right)$$

This step reduces the precision of less important (often high-frequency) coefficients, making many of them zero or small after rounding.

Zigzag Ordering The coefficients are reordered in a zigzag pattern to group low-frequency coefficients together, making it easier to apply thresholding and run-length encoding. Although low-frequency coefficients have lower k values, zigzag ordering ensures that they are contiguous in the 1D array, optimizing run-length encoding.

Run-Length Encoding (RLE) Run-Length Encoding (RLE) compresses sequences of zeros efficiently. After quantization and zigzag reordering, many DCT coefficients will be zero. RLE represents these zero sequences more compactly.

Entropy Coding Finally, the quantized coefficients are encoded using entropy coding techniques like Huffman coding, further reducing the data size.

4 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is a transform used in signal processing and compression, offering advantages over the Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT). The DWT provides a time-frequency representation of the signal, capturing both frequency and location information.

4.1 Overview of DWT

The DWT decomposes a signal into a set of wavelets, which are localized in both time and frequency. This allows for multi-resolution analysis, where different parts of the signal can be analyzed at different scales.

4.2 Key Concepts

- **Wavelets**: Functions that efficiently represent data with sharp changes or edges, localized in time.
- **Scaling and Translation**: Wavelets can be scaled (dilated) and translated (shifted) to capture different frequency components and their locations in the signal.
- **Multi-Resolution Analysis**: DWT performs analysis at multiple resolutions, capturing both coarse and fine details of the signal.

4.3 DWT Algorithm

The DWT of a signal can be computed using recursive filtering and downsampling. The process involves two main steps: decomposition (analysis) and reconstruction (synthesis).

Decomposition (Analysis)

- **Filter Bank**: Apply a pair of filters to the signal: a low-pass filter (L) and a high-pass filter (H). The low-pass filter captures the approximation (low-frequency) components, while the high-pass filter captures the detail (high-frequency) components.
- **Downsampling**: After filtering, the signal is downsampled by a factor of 2 (keeping every other sample) to reduce the data size.
- **Recursive Decomposition**: The decomposition process is recursively applied to the low-pass filtered signal to create a multi-level decomposition.

Reconstruction (Synthesis)

- **Upsampling**: The downsampled components are upsampled by a factor of 2 (inserting zeros between samples).
- **Filter Bank**: Apply the synthesis filters (low-pass and high-pass) to the upsampled components.
- **Combining**: The filtered components are combined to reconstruct the signal.

4.4 Mathematical Formulation

Given a signal $x[n]$:

- **Approximation Coefficients (Low Frequency)**:

$$A_j[k] = \sum_n x[n] \cdot \phi_{j,k}[n]$$

  where $\phi_{j,k}[n]$ are the scaling functions (low-pass).

- **Detail Coefficients (High Frequency)**:

$$D_j[k] = \sum_n x[n] \cdot \psi_{j,k}[n]$$

  where $\psi_{j,k}[n]$ are the wavelet functions (high-pass).

4.5 Advantages of DWT

- **Localization**: Wavelets are localized in both time and frequency, allowing DWT to capture transient features more effectively than DFT or DCT.
- **Multi-Resolution Analysis**: DWT provides a hierarchical representation, enabling analysis at multiple resolutions and scales.
- **Efficient Compression**: DWT often achieves better compression efficiency for images and signals with sharp changes or edges, as it can represent such features more compactly.

4.6 Complexity of DWT

4.6.1 Direct Computation

The direct computation of DWT for a signal of length N involves $O(N)$ operations per level of decomposition. For a full J-level decomposition, the total complexity is:

$$O(N)$$

4.6.2 Fast Algorithms for DWT

Fast DWT algorithms, such as those based on recursive filtering and downsampling, also achieve a complexity of:

$$O(N)$$

These algorithms exploit the hierarchical structure of the wavelet transform to achieve efficient computation.

4.7 Practical Use in Compression

Example: JPEG 2000

JPEG 2000, an advanced image compression standard, uses DWT instead of DCT. Here's a simplified outline of the JPEG 2000 compression process:

- **DWT Computation**: The image is decomposed using DWT into multiple levels of approximation and detail coefficients.
- **Quantization**: The DWT coefficients are quantized to reduce precision and data size.
- **Encoding**: The quantized coefficients are encoded using entropy coding techniques such as arithmetic coding or context-based adaptive binary arithmetic coding (CABAC).

5 Walsh and Hadamard Transforms

The Walsh and Hadamard transforms are discrete, orthogonal, and non-sinusoidal transforms used in various signal processing and communication applications. These transforms are particularly useful for their simplicity and efficiency in hardware implementations.

5.1 Walsh Transform

The Walsh transform is based on Walsh functions, which are square waveforms taking values of +1 or -1. The Walsh functions form an orthogonal basis set for the transform.

Definition The Walsh transform of a sequence $x[n]$ of length N is given by:

$$W[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot \mathrm{wal}(n, k)$$

where $\mathrm{wal}(n, k)$ are the Walsh functions.

Walsh Functions These functions are piecewise constant, switching between +1 and -1. They can be ordered in various ways, such as the Hadamard order or the sequency order (where functions are ordered by the number of zero crossings).

5.2 Hadamard Transform

The Hadamard transform is a specific case of the Walsh transform where the Walsh functions are ordered in the Hadamard order. The Hadamard transform uses Hadamard matrices, which are constructed recursively.

Hadamard Matrix For $N = 2^n$, the Hadamard matrix H_N is defined recursively as:

$$H_{2^n} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix}$$

with the base case $H_1 = [1]$.

Hadamard Transform The Hadamard transform of a sequence x of length N is given by:

$$H(x) = H_N \cdot x$$

where H_N is the Hadamard matrix of order N.

5.3 Properties and Applications

5.3.1 Properties

- **Orthogonality**: Both the Walsh and Hadamard transforms are orthogonal, meaning that their basis functions (Walsh functions or rows of the Hadamard matrix) are orthogonal to each other.
- **Binary Values**: The basis functions take binary values (+1 or -1), making these transforms particularly efficient for digital and hardware implementations.
- **Fast Computation**: The Hadamard transform can be computed in $O(N \log N)$ time using the Fast Walsh-Hadamard Transform (FWHT), similar to the FFT for the DFT.

5.3.2 Applications

- **Digital Communication**: Used in spread-spectrum communication systems, such as CDMA, where Walsh codes are used to separate different users' signals.
- **Image Processing**: Applied in image compression and pattern recognition due to their ability to represent images with binary coefficients.
- **Signal Processing**: Useful in applications requiring fast and efficient transformations, such as real-time signal processing.

6 Comparison and ECG Application

When considering the Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Discrete Fourier Transform (DFT) for compression purposes, each has its own advantages and disadvantages. This section compares these transforms, focusing on compression efficiency, computational complexity, and suitability for various types of signals, including ECG signals.

6.1 Comparison of Transforms

Discrete Wavelet Transform (DWT)

Pros:

- **Multi-Resolution Analysis:** Provides time-frequency representation, capturing both frequency content and its location in time. This is beneficial for non-stationary signals where frequency components change over time.
- **Energy Compaction:** Efficiently compacts energy, especially for signals with sharp changes or transient features. It can represent smooth regions and edges effectively, leading to better compression.
- **Adaptability:** Wavelet basis functions can be chosen to match the characteristics of the signal, offering flexibility in achieving optimal compression.
- **Efficient Compression:** Often results in fewer coefficients with significant values, making it effective for lossy compression schemes.

Cons:

- **Complexity:** More complex to implement and understand compared to DCT. Recursive decomposition and reconstruction add to computational complexity.
- **Compression Artifacts:** Can introduce artifacts, such as ringing or blocking, especially at high compression ratios.

Discrete Cosine Transform (DCT)

Pros:

- **Energy Compaction:** Highly effective in energy compaction for natural images and signals, concentrating most of the signal's energy in a few low-frequency coefficients. This property is exploited in JPEG image compression.
- **Simplicity:** Simpler to implement compared to DWT and DFT. Computationally efficient, especially with fast algorithms like Fast DCT.
- **Widely Used:** Common in image and video compression standards (e.g., JPEG, MPEG) due to its efficiency and ease of implementation.

Cons:

- **Global Transformation:** Operates on fixed-size blocks (e.g., 8x8 blocks in JPEG), which can lead to blocking artifacts in highly compressed images. Does not capture time localization as effectively as DWT.
- **Limited Adaptability:** Uses a fixed basis function (cosine), which may not be optimal for all types of signals, especially those with non-stationary characteristics.

Discrete Fourier Transform (DFT)

Pros:

- **Frequency Analysis:** Provides a complete frequency domain representation of the signal, useful for spectral analysis and filtering.
- **Mathematical Properties:** Well-defined mathematical properties make it suitable for theoretical analysis and certain types of signal processing.

Cons:

- **Complexity:** Computationally intensive ($O(N^2)$) for direct computation. Fast Fourier Transform (FFT) algorithms reduce this to $O(N \log N)$ but are still generally more complex than DCT.
- **Global Transformation:** Assumes the signal is periodic and stationary, which is not ideal for non-stationary signals. Uses complex numbers, increasing the computational and storage requirements.
- **Energy Distribution:** Does not compact energy as efficiently as DCT or DWT, leading to less efficient compression.

6.2 ECG Signal Compression

ECG (Electrocardiogram) signals typically exhibit characteristics that fall between smooth and sharply changing signals. They consist of periodic, repetitive waveforms with distinct features such as P waves, QRS complexes, and T waves.

Characteristics of ECG Signals

- **Periodic Nature:** ECG signals are periodic with well-defined cycles.
- **Sharp Changes:** ECG signals include sharp transitions and spikes (e.g., the QRS complex) as well as smoother segments (e.g., the T wave).

Choice of Transform for ECG Compression

Discrete Wavelet Transform (DWT):

- **Advantages:** The DWT is well-suited for non-stationary signals with transient features, like ECG signals. It captures both the high-frequency spikes and the low-frequency trends, making it effective for representing the signal with fewer coefficients.
- **Energy Compaction:** The multi-resolution analysis of DWT is advantageous for ECG signals, as it allows capturing both the sharp changes and the smooth parts of the signal.

Discrete Cosine Transform (DCT):

- **Advantages:** The DCT is good at capturing periodic and smoothly varying signals but may not be as effective as DWT in capturing the sharp transitions in ECG signals.

Conclusion

The DWT is the preferred choice for compressing ECG signals due to its ability to handle both sharp transitions and smooth components effectively.

7 Conclusions

In this paper, we explored various transforms used for signal compression, specifically focusing on the Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), and the Walsh/Hadamard Transforms. Each of these transforms offers unique advantages and has specific applications where it excels.

7.1 Summary of Findings

- **Discrete Wavelet Transform (DWT):**
  - Provides a time-frequency representation, making it highly suitable for non-stationary signals with transient features.
  - Demonstrates excellent energy compaction, effectively capturing both smooth and sharp changes in signals.
  - Preferred for compressing ECG signals due to its ability to handle both the high-frequency spikes and low-frequency trends in the signal.
- **Discrete Cosine Transform (DCT):**
  - Highly effective for natural images and signals, concentrating most of the signal's energy in a few low-frequency coefficients.
  - Simplifies implementation and is computationally efficient, widely used in image and video compression standards (e.g., JPEG, MPEG).
  - However, it may introduce blocking artifacts in highly compressed images and is less effective for signals with sharp transitions.
- **Discrete Fourier Transform (DFT):**
  - Provides a complete frequency domain representation, useful for spectral analysis and filtering.
  - Computationally intensive and assumes the signal is periodic and stationary, which is not ideal for non-stationary signals.
  - Less efficient for compression compared to DWT and DCT due to moderate energy compaction.
- **Walsh/Hadamard Transforms:**
  - Efficient and simple, particularly suitable for digital and hardware implementations.
  - Useful in applications like digital communication and fast signal processing.
  - Moderate energy compaction compared to DCT and DWT, making it less ideal for natural signals.

7.2 ECG Signal Compression

ECG signals exhibit characteristics of both smooth and sharply changing signals. Given the periodic nature of ECG signals with distinct features such as P waves, QRS complexes, and T waves, the choice of transform for compression is crucial.

- The **Discrete Wavelet Transform (DWT)** is the most suitable for compressing ECG signals. Its multi-resolution analysis capability allows it to effectively capture both the sharp transitions and smooth components of the signal, resulting in efficient compression with minimal loss of critical information.
- While the **Discrete Cosine Transform (DCT)** is good for periodic and smoothly varying signals, it is less effective for capturing the sharp transitions typical of ECG signals.

7.3 Final Remarks

In conclusion, the selection of a transform for signal compression depends on the specific characteristics of the signal and the requirements of the application. The DWT is particularly powerful for non-stationary signals with transient features, making it ideal for ECG signal compression. The DCT remains a robust choice for image and video compression due to its excellent energy compaction for natural signals. Meanwhile, the DFT is more suited for spectral analysis despite its higher computational complexity and less efficient energy compaction. Walsh and Hadamard Transforms offer simplicity and efficiency for hardware implementations but fall short in energy compaction compared to DCT and DWT.

A Karhunen-Loève Transform (KLT)

Consider $x \in \mathbb{C}^N$, a complex-valued, wide-sense stationary signal with mean zero (for simplicity). The covariance matrix of x can be computed numerically:

$$R_x = \mathbb{E}[xx^H],$$

where the superscript H denotes the Hermitian transpose (i.e., $x^H = (x^*)^T$). R_x is real and symmetric, and the eigen-decomposition

$$R_x = Q\Lambda Q^H,$$

gives columns of Q as the eigenvectors of R_x and Λ as a diagonal matrix of the eigenvalues. Q is a unitary matrix, thus $Q^{-1} = Q^H$.

The representation

$$s = Q^H x$$

is known as the Karhunen-Loève Transform (KLT) of x, and we call Q the KLT basis or matrix. The covariance matrix of s is diagonal:

$$R_s = \mathbb{E}[ss^H] = \mathbb{E}[Q^H xx^H Q] = Q^H R_x Q = \Lambda.$$

Thus, the KLT of x results in an uncorrelated representation s, whose covariance matrix has zero cross-correlation terms. In other words, s fully describes x without any statistical redundancy.

Directly cited from:

Gwon, Y., Kung, H. T., & Vlah, D. (2012). Analyzing Interference Effects on the Performance of Heterogeneous Wireless Networks. *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM)*, 5781-5787. Retrieved from https://www.eecs.harvard.edu/ htk/publication/2012-globecom-gwon-kung-vlah.pdf