Journal of Systems Engineering and Electronics (ISSN NO: 1671-1793) Volume 34 ISSUE 11 2024

Mantra video Signal processing using Matlab tool

Gadug Sudhamsu
Department of Computer Science and Engineering,
Jain Deemed to be University

Dr. B S Chandrasekar Shastry
Department of Electronics and Communication Engineering,
Jain Deemed to be University

***Abstract*—An approach to video signal processing using MATLAB, leveraging its capabilities to read, manipulate, and save video frames. The focus lies on a structured, step-wise processing method—referred to here as "mantra"—that applies specific operations to each video frame, enabling repetitive, customizable processing techniques. Using MATLAB's *Computer Vision Toolbox* and built-in functions, we explore techniques including filtering, color adjustment, feature detection, and edge detection. Additionally, methods for assembling processed frames into an output video. The framework supports advanced operations like object tracking, frame stabilization, and spatial transformations, making it versatile for applications in computer vision and multimedia signal processing. Demonstrates MATLAB's effectiveness as a powerful tool for video signal analysis and manipulation, providing a foundation for further research and development in automated video processing.**

Keywords—Video, Signal Processing, Matlab

I. Introduction

The term "multimedia" describes how many media elements—including text, audio, photos, videos, and animations—are combined to create a cohesive digital experience. It blends these diverse media types to provide dynamic and captivating material that can be viewed on a variety of platforms, such as PCs, mobile phones, and the internet. Multimedia aims to improve communication and present information in a more engaging and dynamic manner. The term "multimedia" describes how many media elements—including text, audio, photos, videos, and animations—are combined to create a cohesive digital experience. It blends these diverse media types to provide dynamic and captivating material that can be viewed on a variety of platforms, such as PCs, mobile phones, and the internet. Multimedia aims to improve communication and present information in a more engaging and dynamic manner. Compared to conventional modes of communication that only employ one medium, it can offer users a deeper and more immersive experience by mixing many media aspects. Text makes it possible to communicate concepts and information via written language. By appealing to the auditory senses, audio components including speech, music, and sound effects provide an additional level of communication. Graphics and pictures provide a visual representation and can arouse feelings or explain difficult ideas. While animations produce dynamic and interactive graphics, videos add movement, action, and narrative to multimedia experiences. While animations create dynamic and interactive visuals.

Applications for multimedia can be found in many different sectors and fields. By combining text, graphics, and videos to communicate information, it can help create engaging learning experiences in the classroom. Multimedia is utilised in entertainment to produce immersive virtual reality, video games, and film experiences. Multimedia is used in marketing and advertising to effectively engage consumers and communicate brand messaging. The potential of multimedia has significantly increased due to the development of digital technologies. High-speed internet is widely available, making it simple to access and exchange multimedia information worldwide. Additionally, it is now easier for people and organisations to produce and share their own multimedia material thanks to the advancement of robust software tools and multimedia authoring platforms.

A video signal is a sequence of frames or still images rapidly displayed to create the illusion of motion, encoded into an electrical or digital format for transmission, storage, and playback. This signal plays a central role in delivering moving visual content across various platforms, including television, online streaming, digital cinema, and video conferencing. At its core, a video signal is composed of several key components: frames, resolution, color depth, luminance, and chrominance. The frame rate, measured in frames per second (fps), determines the smoothness of motion, while resolution (like 1080p or 4K) specifies the detail level in each frame.

There are two main types of video signals: analog and digital. Analog video, used in traditional broadcast television formats like NTSC and PAL, encodes visual information as continuous electrical signals. Digital video, which has largely replaced analog, converts images into binary data, allowing for higher quality, greater flexibility in editing, and resilience against noise and degradation. This shift has facilitated the development of compressed video formats, such as MP4 and H.264, which use efficient encoding to reduce file sizes while maintaining high quality.

Processing video signals involves techniques like compression, filtering, and enhancement. Compression reduces the file size for storage and transmission without noticeably affecting quality, while filtering and enhancement can improve image clarity and color balance. Video signals are transmitted via multiple media, including broadcast, internet, and cable, and are displayed on various devices like televisions, monitors, and smartphones. Today, video signals are indispensable to entertainment, education, communication, and surveillance, powering an array of applications from live broadcasting to telemedicine, and continue to evolve with advances in video technology, providing increasingly high-quality and immersive viewing experiences.

Signal processing is a field of study and practice that focuses on analyzing, modifying, and interpreting signals to extract useful information or enhance their quality. Signals can take various forms, such as audio, images, video, biomedical data, radar, and more. By applying mathematical algorithms and techniques, signal processing enables us to manipulate and understand signals for a wide range of applications. Signal processing involves the fundamental tasks of representing, manipulating, and analyzing signals.. A signal is a representation of information that varies over time or space. For example, an audio signal represents variations in air pressure over time, while an image signal represents variations in light intensity over space. Signals can be continuous, meaning they vary smoothly over time or space, or discrete, where they are represented as a sequence of values at specific time or space intervals. Signal processing can be broadly categorized into two domains: analog and digital. Analog signal processing deals with signals in their continuous form and involves using electrical or analog circuits to manipulate and process them. Examples of analog signal processing techniques include analog filters, amplifiers, and modulation techniques. On the other hand, digital signal processing (DSP) focuses on processing signals in their digital or discrete form. Digital signals are represented as sequences of numbers and can be manipulated using computers or digital devices. The advent of powerful computers and digital technology has led to the widespread adoption of digital signal processing techniques. Digital signal processing offers advantages such as flexibility, accuracy, reproducibility, and the ability to handle large amounts of data.

MATLAB (MATrix LABoratory) A high-level programming language and environment explicitly created for numerical and scientific computing is equipped with a robust array of tools and functions tailored for tasks such as data analysis, visualization, algorithm development, and application prototyping. MATLAB's intuitive syntax and extensive libraries make it a popular choice for researchers, engineers, and scientists working in various disciplines. It offers an interactive environment with a command-line interface, often referred to as the MATLAB Command Window. Commands directly in the Command Window and see the results immediately. This interactive nature of MATLAB allows for quick experimentation and exploration of data and algorithms. It provides a simple way to work with variables and data. You can create and manipulate arrays, matrices, and multidimensional data structures using built-in functions or by directly assigning values. It has an extensive collection of mathematical and scientific functions built into the language. These functions cover a wide range of areas, including linear algebra, statistics, optimization, signal processing, image processing, control systems, and more. By using these functions, you can perform complex calculations and analyses with ease. Offers powerful visualization capabilities for creating plots, graphs, and charts. You can plot data, customize the appearance of plots, add labels and annotations, and create 2D or 3D visualizations. MATLAB also supports interactive exploration of plots, allowing you to zoom, pan, and interact with the visualizations. It is not only an interactive environment but also a programming language. You can write scripts and functions to automate repetitive tasks or develop complex algorithms. Supports control flow constructs like loops and conditionals, as well as user-defined functions and modular programming. MATLAB provides various functions for reading and writing data to files. You can import data from common file formats such as CSV, Excel, text files, images, and more. Similarly, you can export data to different file formats for further analysis or sharing with other applications.

It allows you to build standalone applications and user interfaces using its App Designer tool. You can create custom graphical user interfaces (GUIs) with interactive elements, such as buttons, sliders, and plots, without needing extensive knowledge of graphical programming. MATLAB's versatility and vast library of functions make it suitable for a wide range of applications, including data analysis, scientific research, engineering simulations, machine learning, image and signal processing, control systems design, and more. Its user-friendly interface and extensive documentation make it accessible to both beginners and experienced programmers.

A signal pattern refers to a characteristic or repetitive structure present in a signal. Signal patterns can arise from various sources and have different forms depending on the nature of the signal and the underlying phenomenon. Periodic patterns occur when a signal repeats itself over a regular interval. The signal pattern exhibits a consistent shape, amplitude, and frequency content throughout its repetition. Periodic signals encompass various examples, including sine waves, square waves, and sawtooth waves. These patterns are often encountered in applications such as audio signals, periodic vibrations, and periodic electrical signals. Transient patterns occur when a signal exhibits a specific behavior or shape for a finite duration and then returns to its original state. Transients can be caused by abrupt changes, events, or disturbances in the signal. Examples include sudden spikes or pulses in an electrical signal, the initial response of a system to an input, or the attack portion of a musical note. Transient patterns are often of interest in signal analysis to understand the response of systems or detect specific events. Random patterns refer to signals that do not exhibit a discernible repetitive structure or predictable behavior. Random patterns can arise from various sources, such as noise in electrical signals, fluctuations in natural phenomena, or random processes. Analyzing random patterns often involves statistical methods and techniques to characterize the statistical properties of the signal, such as probability distributions or correlations. Modulated patterns occur when a signal carries or modulates another signal. Modulation entails altering specific attributes of a carrier signal, such as amplitude, frequency, or phase, in accordance with the characteristics of the modulating signal. Amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM) are among the most common modulation techniques used in various communication systems. Modulated patterns are used extensively in communication systems, where information is encoded and transmitted using modulated signals.

II. Literature survey

Meng, Qing-Xiang, et al. [1] An open-source MATLAB package has been developed to facilitate the generation of numerical models for Finite Element Method (FEM) and Discrete Element Method (DEM) simulations based on digital images. This package is particularly valuable for researchers and engineers who analyze the mechanical behavior of heterogeneous materials with complex internal structures. By supporting both 2D bitmap and 3D volumetric images, this tool can transform digital representations of materials into accurate computational models, opening new possibilities in various fields, including material science, biomechanics, geotechnics, and engineering. One of the package's core features is its ability to process and vectorize digital images to create detailed numerical models. It includes a variety of geometrical and analytical routines designed for handling bitmap images. These routines can vectorize 2D images, compute geometric information, and transform this data into useful statistical metrics, essential for setting up accurate simulations. With these capabilities, users can analyze geometric properties, track variations within complex structures, and model the real-world microstructure of heterogeneous materials. For instance, this can be useful for materials like composite materials, soils, biological tissues, or any material that has distinct regions or phases within its structure.

The package's support for complex, connected regions is a notable advantage, as it enables the modeling of materials with intricate internal connectivity, such as composite or porous materials. In conventional modeling techniques, handling connected regions or heterogeneous materials with multiple phases is challenging and often requires simplification. This MATLAB package allows users to model these materials more faithfully, preserving the complexity of their microstructure and enabling more accurate simulations of their mechanical behavior under stress, strain, or other applied forces. This fidelity makes it possible to obtain insights into the performance and failure mechanisms of materials under different conditions, which is critical for design, safety, and optimization purposes.

Additionally, the package includes analytical tools that allow users to calculate various geometric statistics for each region within an image. These statistics provide valuable information about the distribution and organization of different phases in heterogeneous materials. For instance, one can evaluate metrics like area, volume, connectivity, and orientation, which can influence the material's mechanical properties. These detailed metrics support not only a comprehensive material analysis but also allow for targeted adjustments within the model, facilitating iterative design and refinement in engineering processes.

Four case studies demonstrate the package's robustness, efficiency, and ease of use. These examples highlight the package's application in creating models for complex materials, handling large data sets, and generating reliable results efficiently. Users without extensive experience in FEM or DEM modeling can still effectively apply this package due to its user-friendly design and clear documentation. The open-source nature of this MATLAB package further increases its accessibility, enabling community-driven improvements, customization, and broad adaptability across research domains.

As a ready-to-use software package, it holds considerable potential for advancing the analysis of heterogeneous materials in various applications. By bridging the gap between digital imaging and computational modeling, this tool can streamline workflows for scientists and engineers, reduce modeling time, and improve accuracy. Its open-source licensing also makes it an attractive choice for academic and industry professionals who seek cost-effective, customizable solutions for material analysis. In summary, this MATLAB package represents a robust, versatile tool for advancing FEM and DEM applications, providing researchers with powerful capabilities to study and simulate the behavior of complex materials in unprecedented detail.

Wani, Lalita K., Dhananjay E. Upasani, and Anupama Deshpande, et al. [2] This research paper delves into the scientific study of the sacred sound "OM" (or "AUM") and its profound effects on human mental and physical health, with an emphasis on frequency analysis and the variations in its chanting patterns. Historically, sounds like "OM," the Gayatri Mantra, and Mrityunjaya Mantra have been significant in various cultural, spiritual, and health-related contexts, as indicated by ancient texts and practices. Manuscripts and studies suggest that these sacred sounds resonate with frequencies that influence the human body and mind positively. While scientific support is growing, more comprehensive research is still needed to explore and verify the impact of these sounds, especially "OM," on critical bodily functions such as heart health. In Hindu and yogic literature, particularly the Mandukya Upanishad, "OM" or "AUM" is described as a syllable with unique vibrations that symbolize different aspects of consciousness and existence. The sound "OM" is traditionally divided into three parts: the sounds "A," "U," and "M." Each part holds symbolic meanings and is said to resonate with specific parts of the human body, creating physical and mental harmony. However, due to the variety of interpretations in ancient texts, finding a universally accepted or ideal method for chanting "OM" can be confusing, especially for individuals without a deep understanding of these traditions.

A. Prochazka, O. Vysata and V. Marik, et al [3] Digital signal processing plays a crucial role in the management and analysis of digital information. In educational settings, DSP techniques can be applied to process and analyze various forms of data, including audio, video, and sensor data. For instance, DSP can enhance the quality of educational videos, optimize audio content for clarity, and analyze speech patterns to support language learning. Furthermore, through the use of signal processing algorithms, educators can extract meaningful insights from vast amounts of educational data, enabling informed decision-making and targeted instructional strategies.

The integration of CI and DSP in education facilitates the development of innovative educational tools and applications. For example, intelligent tutoring systems (ITS) leverage CI techniques to provide personalized instruction, while DSP can improve the interactive capabilities of these systems by enabling real-time feedback and assessment. Such systems can adapt to the learning pace and style of each student, thereby enhancing engagement and motivation. Additionally, these technologies can support collaborative learning environments by providing tools for peer assessment, discussion, and feedback.

Emerging technologies, such as augmented reality (AR) and virtual reality (VR), further enhance the educational landscape when integrated with CI and DSP. These technologies create immersive learning environments that stimulate student engagement and understanding. For example, VR simulations can provide experiential learning opportunities in subjects like science and engineering, while CI algorithms can analyze student interactions within these environments to assess learning outcomes and adjust instructional strategies accordingly. DSP can also improve the quality of AR and VR content by processing and rendering high-quality audio-visual experiences. Mathematical tools play a fundamental role in both CI and DSP applications in education. Techniques such as statistical analysis, linear algebra, and calculus are essential for developing and implementing algorithms that drive intelligent educational technologies. Educators can leverage these mathematical concepts to teach students critical thinking and problem-solving skills, empowering them to engage with complex computational and data-driven challenges. Moreover, integrating mathematical tools with CI and DSP can enhance the curriculum by providing practical applications that resonate with students.

Savioja, Lauri, Vesa Välimäki, and Julius O. Smith et, al [4] Graphics processing units (GPUs) in vidoe signal processing. While GPUs were originally designed for visual displays, they have proven effective in audio applications as well. The text presents case studies demonstrating the advantages of GPU implementation in tasks such as additive synthesis, where real-time computation of millions of sine waves is achieved with significantly improved performance compared to CPU-based implementations. Additionally, the use of GPUs has shown faster processing speeds in tasks involving Fast Fourier Transforms (FFTs) and filters. Overall, the text emphasizes the potential of GPUs in enhancing the efficiency and performance of audio signal processing algorithms.

Zeng, Yuni, et al [6] In video classification tasks, there exist inherent relationships between various tasks, such as speakers' accent and identification. To leverage these relationships, the authors propose a Deep Neural Network (DNN)-based multi-task model named Gated Residual Networks (GResNets). GResNets combine Deep Residual Networks (ResNets) with a gate mechanism, enhancing representation extraction in comparison to Convolutional Neural Networks (CNNs). In their approach, the authors replace two feed-forward convolution layers in ResNets with two multiplied convolutional layers. Experimental results demonstrate that the GResNets multi-task model outperforms task-specific models that are trained independently, achieving higher accuracy.

M. Mauch, S. Ewert et al [7] An introduces a toolbox with modules for simulating various types of audio degradations, including additive noise, channel effects, quantization, and time-scale modifications. The toolbox facilitates the evaluation of audio processing algorithms and systems by introducing specific degradations into audio signals. The article emphasizes the importance of robustness evaluation in applications like speech recognition and audio coding, considering real-world challenges. Several case studies demonstrate the toolbox's effectiveness in evaluating algorithm performance under different degradations. The article concludes that the Audio Degradation Toolbox is a valuable tool for assessing the robustness and reliability of audio systems, providing researchers and practitioners with the means to evaluate algorithms under realistic conditions.

III. System Requirements

TABLE I. SYSTEM DETAILS

Sl. No.	Particulars	Description Matlab
1	Operating System	Windows 11 Windows 10 (version 20H2 or higher) Windows Server 2019 Windows Server 2022
2	Processor	Minimum: Any Intel or AMD x86–64 processor. Recommended: Any Intel or AMD x86–64 processor with four logical cores and AVX2 instruction set support. Note: A future release of MATLAB will require a processor with AVX2 instruction set support.
3	RAM	Minimum: 4 GB Recommended: 8 GB
4	Storage	3.8 GB for just MATLAB 4-6 GB for a typical installation 23 GB for an all products installation. An SSD is strongly recommended.
5	Graphics	No specific graphics card is required, but a hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended. GPU acceleration using Parallel Computing Toolbox requires a GPU with a specific range of compute capability.

IV. Research Methodology

The research methodology for analyzing mantra video signal processing using MATLAB involves a structured approach that combines various stages of data acquisition, processing, analysis, and interpretation. This methodology is designed to explore the effects of sacred sounds, such as mantras, on both audio and visual signals within video recordings.

A. Data Acquisition:

The first step involves selecting a suitable mantra video for analysis, ensuring it captures the chanting of a mantra under controlled conditions. The video should be in a high-quality format to facilitate accurate processing. A video file, such as a recording of the Gayatri Mantra, is chosen, and metadata, including frame rate and duration, is noted.

B. Preprocessing:

Once the video is obtained, MATLAB is used to read the video file and extract frames. This process includes converting the frames to grayscale for easier processing and reducing computational complexity. Video frames are analyzed individually to focus on the visual representation of the chanting. Concurrently, audio processing techniques are applied to extract the sound from the video. This sound is analyzed to isolate the frequencies associated with the mantra chanting.

C. Signal Processing:

Apply filters to the signal to reduce undesired noise or change its frequency makeup (e.g., high-pass, low-pass, bandpass filters). Adjust the audio's frequency response by equalisation. Dynamics processing: Adapt the audio signal's dynamic range using strategies including compression, expansion, or limitation.

D. Processing in the frequency domain

Fourier Transform: Use methods like the Fast Fourier Transform (FFT) to transform the audio signal from the time domain to the frequency domain. Analyse the signal's frequency content using a spectral analysis technique to pinpoint specific frequency components or features. Utilise methods like filtering, spectral shaping, or spectral effects to alter the frequency components.

E. Impacts and Improvements:

Reverberation: By incorporating reflections and decay into the audio input, this effect simulates the acoustic environment of a space. To produce echoes or spatial effects, add temporal delays between audio streams.
Modulation: Use effects like chorus, flanging, or phasing to give the audio a variety of textures and motion.

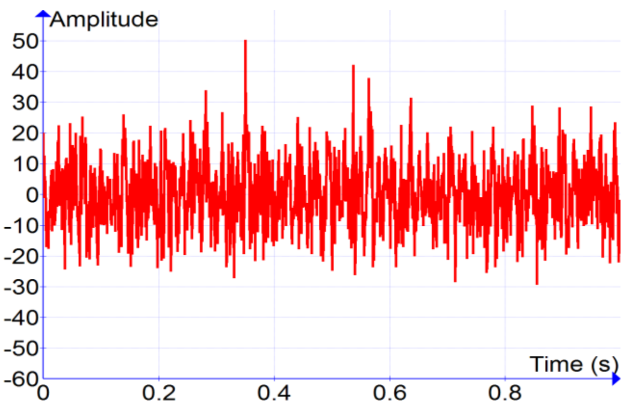


Fig. 1. Sample signal processing

The discrete Fourier transform (DFT) of a series of complex or real numbers can be quickly computed using the Fast Fourier Transform (FFT) algorithm. It reveals information about the frequency content of a signal by breaking down a signal into its individual frequency components.

F. Discrete Fourier Transform (DFT):

The Discrete Fourier Transform (DFT) is a fundamental mathematical tool used in signal processing, image analysis, communication systems, and many other areas of science and engineering. It enables the analysis and manipulation of discrete signals or sequences in the frequency domain, providing valuable insights into their frequency components.

The DFT is a discrete version of the Fourier Transform, which was introduced by French mathematician Joseph Fourier in the early 19th century. The Fourier Transform is a mathematical technique that decomposes a continuous signal into its constituent frequencies, revealing the amplitude and phase of each frequency component. The DFT extends this concept to discrete signals, which are sequences of sampled values at evenly spaced intervals.

Given a sequence of N discrete samples x[n], where n = 0, 1, ..., N-1, the DFT computes a set of complex numbers X[k], where k = 0, 1, ..., N-1. Each X[k] represents the amplitude and phase of the corresponding frequency component in the original signal. The DFT equation is defined as follows:

$$X[k] = \sum[n=0 \text{ to } N-1]\ x[n] * e^{\wedge}(-j2\pi n*k/N) \quad (1)$$

Where:

X[k] is the kth frequency component of the transformed signal.
x[n] is the nth sample of the input signal.
N is the total number of samples in the input signal.

N complex values X[k], where each X[k] corresponds to a different frequency bin ranging from 0 Hz to (N-1) Hz. The magnitude of each X[k] represents the amplitude of the corresponding frequency component, while its argument gives the phase angle.

In this paper we are interested in extracting the harmonious. The fundamental frequency can also be determined.

The DFT algorithm can be efficiently computed using the Fast Fourier Transform (FFT) algorithm, which significantly reduces the number of operations needed for large values of N. The FFT has various implementations, such as the Cooley-Tukey algorithm and the Radix-2 algorithm, which exploit symmetries and periodicities in the DFT computation to speed up the process.

The DFT has numerous applications in different fields. In signal processing, it is used for spectral analysis, filtering, and feature extraction. In audio processing, the DFT is utilized for audio compression, equalization, and sound synthesis. In audio signal analysis, the two-dimensional DFT is employed for audio compression and pattern recognition. Additionally, the DFT plays a crucial role in digital communication systems, where it is used for channel estimation, equalization, and modulation/demodulation.

The computation complexity is O(N^2), making it impractical for real-time processing or large datasets.

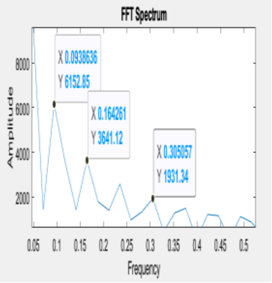
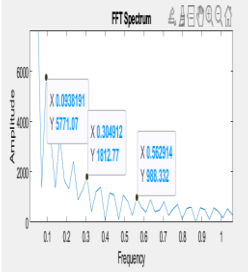
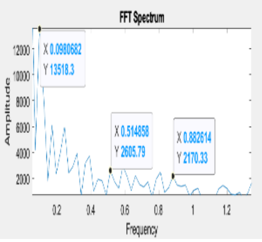
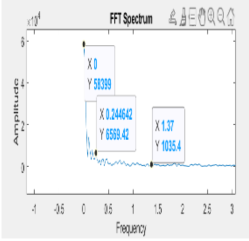
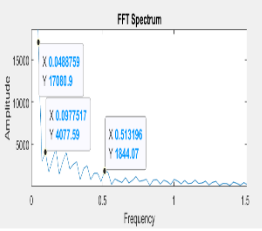
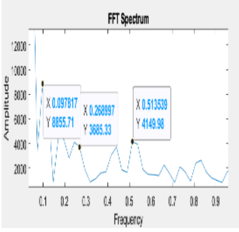
	Matlab 30 sec	Matlab 60 sec
Fig 1		
Fig 2		
Fig 3		

Fig. 2. Amplitude and Frequency domain of video signal processing

However, with the advent of the FFT, this limitation is mitigated, and the DFT becomes computationally feasible for many practical applications. In conclusion, the Discrete Fourier Transform is a powerful mathematical tool for analyzing and processing discrete signals in the frequency domain. Its ability to reveal frequency components and their associated amplitudes and phases makes it a cornerstone of modern signal processing and numerous other scientific and engineering disciplines. The FFT has further enhanced its utility, enabling efficient computation and expanding its range of applications to diverse fields, thereby shaping the way we analyze and manipulate digital signals.

The FFT makes the DFT useful for real-time and fast signal processing by lowering its computational complexity from O(N2) to O(N log N). The FFT is an essential tool in many audio and signal processing applications because it allows for tasks like spectral analysis, filtering, and audio signal creation by translating signals from the time domain to the frequency domain.

V. MAT LAB FOR SIGNAL PROCESSING

The provided code snippet demonstrates the process of recording audio using the MATLAB audio recorder function, visualizing the audio signal in both the time and frequency domains, and saving the recorded audio as a WAV file

Four speech Samples have been used for experimentation. This is a conversation, a speech, or any spoken content. It will provide a diverse range of frequencies and amplitudes.

Instrumental piece as the audio signal. Music contains a wide variety of frequencies and complex waveforms, which can produce interesting patterns in the time and frequency domains. Generate a white noise signal using a random number generator. White noise contains all frequencies at equal intensity and can be used for testing purposes or as a source of background noise. Sine Wave: Generate a pure sine wave of a specific frequency, such as 440 Hz (A4), using a sinusoidal function. This will produce a simple and regular waveform with a single frequency component.

VI. RESULTS

The x-axis is labeled as "Frequency (Hz)," the y-axis is labeled as "Amplitude," and the title of the subplot is designated as "Frequency Domain Plot of the Video Signal."

The time domain plot in the program visually depicts the changes in amplitude of the video signal (x) over time. The x-axis denotes time in seconds, while the y-axis indicates the signal's amplitude. By displaying the audio signal in the time domain, one can observe the variations and patterns within the waveform. The peaks and valleys in the plot reflect fluctuations in the signal's amplitude, and the overall shape of the waveform represents the characteristics of the video signal.

This study investigates the effects of mantra chanting through the analysis of video signal processing using MATLAB, aiming to uncover insights into the auditory and visual dimensions of this ancient practice. Mantras, such as the Gayatri Mantra, are deeply rooted in cultural traditions and have been associated with mental and physical well-being for centuries. The research aims to explore how these sacred sounds impact individuals, focusing on both the audio and visual components of mantra chanting. The methodology begins with data acquisition, wherein a high-quality video of a mantra chant is selected for analysis. The MATLAB environment is utilized to process the video, employing functions like VideoReader to extract frames and audio data. Preprocessing includes converting the video frames to grayscale, facilitating easier visual analysis, while the audio signal undergoes Fast Fourier Transform (FFT) to analyze its frequency content.

The core of the analysis comprises audio signal processing and visual signal processing. The audio signal is subjected to frequency domain analysis using FFT, revealing distinct peaks that correspond to the fundamental frequency and harmonics of the mantra. This spectral analysis provides insights into the acoustic properties of the chanting, indicating which frequencies dominate the sound profile. Additionally, a time domain analysis is performed to present the waveform of the audio signal, allowing the identification of rhythmic patterns and variations in amplitude over time. In parallel, the visual analysis of the video frames extracts key features, such as brightness and motion, with techniques like edge detection highlighting facial expressions and gestures during the chanting. This visual data serves to offer insights into the emotional and physical state of the practitioner while chanting.

A significant aspect of the research is the correlation analysis between the audio and visual signals. By aligning the audio frequency characteristics with visual features, the study examines how the sound of the mantra affects visual representations. For instance, it explores whether significant audio peaks correspond to observable changes in facial expressions or body posture. The results reveal several key findings: the FFT analysis indicates that the mantra audio exhibits dominant frequencies, suggesting the acoustic properties that may influence the effectiveness of chanting in promoting well-being. Observations from the video frames show that visual features change during the chanting process, potentially reflecting a state of relaxation or meditation. The analysis indicates that visual stability, such as smoother transitions in brightness, correlates with the calming effects associated with mantra chanting.

Moreover, the correlation between audio characteristics and visual responses indicates that both modalities contribute to the experience of chanting. The study interprets these findings as evidence of the holistic impact of mantra chanting, supporting the hypothesis that the sound of the mantra significantly influences the emotional state of the practitioner. In conclusion, the analysis of mantra video signal processing using MATLAB provides a comprehensive understanding of the interplay between sound and visual expression in mantra chanting. The study underscores the significance of combining audio and visual data to assess the effects of sacred sounds on well-being. Future research could build upon these findings by exploring different mantras and larger datasets to further validate the health benefits associated with such practices. Ultimately, this research contributes to a growing body of knowledge that seeks to bridge the gap between traditional practices and scientific inquiry, offering a potential pathway for integrating ancient wisdom with modern scientific methods.

VII. Conclusion

This study successfully utilized video signal processing in MATLAB to explore the effects of mantra chanting on human well-being, highlighting the interconnectedness of auditory and visual elements. The audio analysis revealed distinct frequency characteristics through Fast Fourier Transform (FFT), identifying dominant frequencies that may enhance the benefits of chanting, such as relaxation and emotional stability. The time domain analysis further illustrated rhythmic patterns in the audio, enriching the understanding of the chanting experience.

On the visual side, the examination of video frames uncovered key features like facial expressions and gestures, which reflected the practitioner's emotional states during chanting. Notably, the correlation between audio peaks and visual changes emphasized how sound influences visual perception, suggesting a synergistic effect that amplifies the overall impact of mantra practice.

The findings suggest practical applications in wellness and therapeutic settings, promoting relaxation and emotional well-being. Future research should explore a wider variety of mantras, longer chanting durations, and diverse participant demographics to further validate these findings. Overall, this study bridges traditional practices with modern technology, paving the way for deeper insights into the profound effects of sacred sounds on mental health and emotional resilience.

References

[1] Meng, Qing-Xiang, et al. "DigiSim—an open source software package for heterogeneous material modeling based on digital image processing." Advances in Engineering Software 148 (2020): 102836.

[2] Wani, Lalita K., Dhananjay E. Upasani, and Anupama Deshpande. "Analysis Of Om (AUM) Chanting Using Machine Learning." Educational Administration: Theory and Practice 30.4 (2024): 220-232.

[3] A. Prochazka, O. Vysata and V. Marik, "Integrating the Role of Computational Intelligence and Digital Signal Processing in Education: Emerging Technologies and Mathematical Tools," in *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 154-162, May 2021, doi: 10.1109/MSP.2021.3058634. keywords: {Collaboration;Digital signal processing;Speech recognition;Software tools;Systems modeling;Mathematics;Educational courses;Curriculum development;Engineering education},

[4] Giraldo, Jhony H., et al. "The emerging field of graph signal processing for moving object segmentation." International workshop on frontiers of computer vision. Cham: Springer International Publishing, 2021.

[5] Hadad, Elior, et al. "Multichannel audio database in various acoustic environments." *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2014.

[6] Kim, Byung-Gyu, and Dong-San Jun. "Artificial Intelligence for Multimedia Signal Processing." *Applied Sciences* 12.15 (2022): 7358.

[7] Kellermann, Walter, Rainer Martin, and Nobutaka Ono. "Signal processing and machine learning for speech and audio in acoustic sensor networks." *EURASIP Journal on Audio, Speech, and Music Processing* 2023.1 (2023): 54.

[8] Spanias, Andreas, Ted Painter, and Venkatraman Atti. *Audio signal processing and coding*. John Wiley & Sons, 2006.

[9] Giannakopoulos, Theodoros. "pyaudioanalysis: An open-source python library for audio signal analysis." *PloS one* 10.12 (2015): e0144610.

[10] Cobos, Maximo, and Sandra Roger. "SART3D: A MATLAB toolbox for spatial audio and signal processing education." *Computer Applications in Engineering Education* 27.4 (2019): 971-985.

[11] .Zeng, Yuni, et al. "Spectrogram based multi-task audio classification." Multimedia Tools and Applications 78 (2019): 3705-3722.

[12] Saadi, Slami, Ahmed Merrad, and Ali Benziane. "Novel secured scheme for blind audio/speech norm-space watermarking by Arnold algorithm." Signal Processing 154 (2019): 74-86.