Quantifying Interpersonal Synchrony: A Multi-Modal Approach Using Hand Movement Analysis

Dr Roi Yozevitch, Zeev Fischer, Elhai Agassi, Danielle Musai

Abstract

This study investigates two methods for real-time hand tracking and gesture recognition in video conferencing: the Leap Motion Controller (LMC) and MediaPipe Hands integrated with WebRTC [1, 2]. We researched both approaches to explore their potential applications in enhancing non-verbal communication in digital spaces. The Leap Motion Controller, known for its sub-millimeter accuracy in 3D space, has been widely applied in fields such as human-computer interaction, robotics, and gesture recognition [3].It directly computes hand position and orientation using key points, offering precise data for hand movement analysis [4]. Despite its limited field of view compared to depth cameras like Kinect, the LMC's accuracy makes it valuable for detailed hand gesture studies [5]. We also researched the integration of MediaPipe Hands, an open-source machine learning solution for hand tracking created by Google, with WebRTC tools for real-time video communication [6]. This approach allows for hand gesture recognition directly from standard webcam feeds, potentially making it more accessible and scalable for widespread video conferencing use [7]. Our research explores the capabilities of both the LMC and the MediaPipe-WebRTC integration, providing insights into the feasibility and effectiveness of real-time hand movement synchronization analysis in web-based video conferencing [8].The findings have implications for enhancing non-verbal communication in digital spaces and developing more intuitive video conferencing tools [9].

1 Introduction

Synchronization is a fundamental phenomenon observed across various scales in nature [10]. In human interactions, synchronization manifests as the unconscious mimicry of postures, gestures, and vocal patterns, a process known as interpersonal synchrony [11]. This synchrony plays a crucial role in our social lives, fostering feelings of connection, enhancing communication effectiveness, and contributing to the development of rapport and empathy [12].

The importance of synchronization extends beyond mere mimicry; it serves as a non-verbal indicator of social cohesion and mutual understanding. In face-to-face interactions, synchronization of body movements, including hand gestures, has been linked to increased liking, smoother conversations, and even improved learning outcomes in educational settings [13]. Moreover, the ability to synchronize with others is considered a key component of social cognition, with implications for cognitive development and social bonding throughout the lifespan [14].

In recent years, the advent of video conferencing technologies has revolutionized remote communication, particularly in the wake of global events necessitating social distancing [15]. While these platforms have become indispensable for maintaining personal and professional connections, they often lack the nuanced interpersonal cues present in face-to-face interactions [16]. One such crucial element is the synchronization of hand movements between conversing parties, which plays a significant role in non-verbal communication and social bonding [17].

As our reliance on digital communication increases, understanding how synchronization manifests in these mediated environments becomes increasingly important. Recent advancements in computer vision and machine learning have opened new avenues for analyzing human behavior in digital contexts [18]. Our study aims to investigate the degree of hand movement synchronization between individuals during video calls, and to explore how this synchronization relates to various aspects of communication quality and social connection. By overlaying hand landmarks on video streams and logging movement data, we provide a unique platform for both real-time visual feedback and post-hoc analysis of non-verbal cues in digital interactions [7].

This research contributes to the growing body of work on computer-mediated communication and social signal processing. It offers insights into how technology can be used to enhance our understanding of non-verbal communication in digital spaces, potentially informing the development of more intuitive and socially aware video conferencing tools [19].

2 Related Works

Our research builds upon several key areas of previous work in interpersonal synchrony while introducing novel approaches to address limitations in existing methodologies. The mirror game has been a fundamental paradigm in studying interpersonal synchrony, where two players mimic each other's hand movements along a one-dimensional track. This seminal work provided valuable insights into the dynamics of joint improvisation and synchronization. Subsequent studies extended this concept to three-dimensional movements, increasing ecological validity. Our study builds upon this foundation by adapting the mirror game concept to a digital environment. Instead of physical tracks or specialized equipment, we utilize the Leap Motion Controller and MediaPipe to capture free-form hand movements in three-dimensional space. This approach maintains the essence of the mirror game—studying synchronous movements between individuals—while making it applicable to remote interactions, opening new avenues for understanding synchrony in modern, technologically mediated social interactions.

Previous research has also employed various technologies to capture synchronous movements, ranging from specialized motion capture systems to wearable devices like wrist-worn accelerometers. While these methods have their merits, they often require specialized equipment or offer limited information about fine hand movements. Our approach utilizes the Leap Motion Controller for high-precision tracking and MediaPipe for accessibility, striking a balance between accuracy and practicality. This multi-modal approach allows for detailed capture of hand postures and movements, offering a richer dataset for analysis compared to previous methods.

In contrast, some researchers have used video-based methods, such as Motion Energy Analysis (MEA), to assess nonverbal synchrony. These techniques, while non-invasive, often lack the precision in tracking specific body parts that our approach provides. Our integration of MediaPipe Hands with WebRTC advances this field by allowing for detailed hand gesture recognition directly from standard webcam feeds, making it more accessible and scalable for widespread video conferencing use.

Moreover, much of the existing research on interpersonal synchrony has focused on in-person interactions. Our work specifically addresses synchrony in remote video conferencing scenarios, which is particularly relevant given the increased reliance on virtual communication in recent years. This focus on remote settings distinguishes our study and provides insights into how synchrony manifests in digital spaces.

While previous studies have typically employed a single method for analyzing synchronization, our research is unique in directly comparing different methods—DTW, CCA, Cross-Correlation, and Wavelet Coherence—within the same experimental framework. This comparative approach provides valuable insights into the strengths and limitations of each method in the context of hand movement synchronization.

By addressing these key areas and introducing novel elements, particularly the integration of MediaPipe with WebRTC, our study contributes to a more comprehensive understanding of interpersonal synchrony in digital spaces. This approach not only builds upon existing research but also paves the way for more accessible and applicable synchrony analysis in everyday video conferencing scenarios.

3 Synchronization Analysis Methods

As mentioned above, we took two different approaches to try and analyze synchronization.

3.1 the Leap Motion Controller (LMC)

The first approach is the Leap Motion Controller (LMC) device, which is increasingly used in various fields such as human-computer interaction, robotics, computer gaming, automatic sign language interpretation, and more. Hand gesture recognition, in particular, has become a growing focus of attention due to the LMC's advanced capabilities [20, 21]. The Leap Motion Controller has opened up new opportunities in hand gesture recognition by enabling precise and intuitive hand tracking in 3D space [22].

The LMC is specifically designed for hand gesture recognition, where it directly computes the position and orientation of the hands using key points, such as joints and fingers. This allows for the real-time detection and analysis of detailed hand movements. Although the amount of data captured is somewhat limited compared to other depth-sensing cameras like Microsoft's Kinect, the Leap Motion Controller compensates with exceptional accuracy, offering sub-millimeter precision [23]. This high level of detail makes it ideal for tasks requiring fine motor control, where other sensors might fall short.

Unlike traditional multi-touch solutions that require direct contact with a surface, the Leap Motion Controller is an above-surface sensor, designed for use in realistic stereo 3D interaction systems. This enables users to interact naturally with virtual environments, making it a key component in applications such as virtual and augmented reality [24, 25]. Its ability to track multiple hands simultaneously allows for complex interactions, where both hands can be used to manipulate virtual objects or perform tasks in a 3D space.

While other systems may provide broader functionality, such as full-body tracking or environmental sensing, the Leap Motion Controller excels in hand-based interactions. Its focus on hand gestures and precise tracking makes it particularly useful in scenarios where fine detail and accuracy are crucial, such as virtual surgical training, sign language interpretation, or creative design interfaces [26, 27]. Figure 1 shows both the Leap Motion Controller and a 3D output of two hands being tracked in real time, illustrating the device's capabilities for capturing complex, simultaneous hand movements.

Figure 1: Two hands being tracked in real-time using the Leap Motion Controller.

3.2 Mediapipe

The second approach is the built-in Python library, MediaPipe. MediaPipe Hands, a robust and versatile Python library developed by Google for real-time hand tracking and gesture recognition. MediaPipe stands out due to its efficient use of machine learning models to detect and track hand landmarks directly from video feeds, making it suitable for a variety of applications, including video conferencing, virtual reality, and gesture-based control systems [6].

One of the primary advantages of MediaPipe Hands is its ability to work with standard 2D webcam footage, eliminating the need for specialized hardware such as depth cameras or dedicated tracking devices. This makes it more accessible for a broader range of users and applications, particularly in settings where cost and setup simplicity are critical. The ability to process hand tracking from a regular webcam feed is a significant advantage for remote communication platforms, as it allows for seamless integration without the need for additional equipment [2].

In our project, MediaPipe Hands was integrated with WebRTC, a powerful tool for real-time communication, enabling us to overlay hand landmarks on live video streams. This approach allows for dynamic analysis of hand movements during video calls, providing a non-intrusive means of studying hand gesture synchronization. Furthermore, the use of machine learning models in MediaPipe ensures robust performance across various lighting conditions and backgrounds, making it a reliable solution for real-world usage scenarios.

4 Workspace

As we are using two different synchronization analysis methods, two different workspaces were required.

4.1 Remote Video Conference Synchronization System

Our first approach to solving the synchronization analysis problem was to leverage an existing, widely-used video conferencing platform. The plan was to access the raw video data during calls, perform real-time body movement analysis, and display the results within the conversation interface. After surveying various platforms, we identified Zoom as the most promising candidate due to its extensive SDK for developer plugins, particularly for desktop applications. Initially, Zoom's SDK appeared promising, offering access to numerous parameters and features. However, we discovered that while the platform provided various customization options, it did not allow direct access to the raw video data stream through its standard SDK. This functionality was only available through Zoom's separate streaming service, which required building a standalone application rather than integrating with the official Zoom client. Zoom provided a C++ demo application that could be compiled and packaged through Visual Studio while using their streaming service. However, this approach ultimately proved unsuccessful despite significant time and resource investment. The supporting mechanism for running this software was highly complex, and various modifications led to unexpected issues that required extensive troubleshooting. The challenges of maintaining and debugging this system became prohibitively resource-intensive. These complications led us to abandon this approach in favor of developing our own streamlined mechanism from the ground up, focusing on simplicity and direct control over the entire process.

Our approach integrates MediaPipe Hands with WebRTC to create a web-based video conferencing system for real-time hand tracking and synchronization analysis. The front end uses HTML to display local and remote video streams with basic controls. The backend is built on Node.js and Express, using Socket.IO for real-time communication.

The server manages WebSocket connections, handling events like 'offer', 'answer', and 'ice-candidate' for WebRTC peer-to-peer connections. This architecture enables low-latency video conferencing with integrated hand-tracking capabilities.

Combining WebRTC with MediaPipe Hands allows us to analyze hand movements during video calls without additional hardware. The system is designed for easy deployment, running on the Heroku platform. This implementation provides a foundation for studying interpersonal synchrony in remote interactions using web technologies and computer vision in contrast to Leap which required a local meeting.

4.2 Unity Workspace

Due to unprecedented regional circumstances and resulting operational constraints, we needed to revise our initial research scope. This led us to pivot towards a more focused approach using locally controlled data collection methods. Specifically, we redirected our investigation to analyze synchronization algorithms using the Leap Motion Controller as a controlled, stationary data source. This modification allowed us to maintain research continuity while adapting to the practical limitations of our environment.

The Leap Motion Controller is a device that tracks hand and finger movements in 3D space. It uses infrared cameras and LEDs to detect and analyze hand positions, gestures, and movements with high precision. When integrated with Unity, the Leap Motion Controller can be used to interact with 3D objects and environments in real time through natural hand movements. This allows developers to create immersive experiences where users can interact without traditional input devices like keyboards or game controllers [23, 28].

To use Leap we used Unity to edit its commands. Unity is a popular real-time 3D development platform used for creating video games, simulations, virtual reality (VR), and augmented reality (AR) applications. It provides an intuitive environment for developers to build interactive 3D experiences, with support for physics, rendering, audio, animation, and more. Unity can integrate with various hardware devices, such as the Leap Motion Controller, to enhance interactivity, especially for motion tracking and gesture recognition [29, 30].

5 Experiment Recordings

To analyze the efficiency of the four algorithms we chose to use regardless of the platform used Unity with the Leap Motion Controller or the Video Conference we created using MediaPipe we will look at four different scenarios.

5.1 Synchronized Movement

Synchronized movement is where two people will move their hands in a pattern known to both participants. We will use this to create a baseline for our analysis. As two hands can never move exactly the same we do not expect to ever get 100% accuracy however we are expecting to get the highest and best results in this case.

5.2 Unsynchronized Movement

Unsynchronized movement where again two people will move their hands in 2 different patterns there will not be any specific pattern and we do not care if there is a moment of synchrony as this is natural in two people that are looking at each other. As with perfection, there can not be 100% chaos; we do not expect to get 0% synchrony; however, we do expect to get a very low percentage.

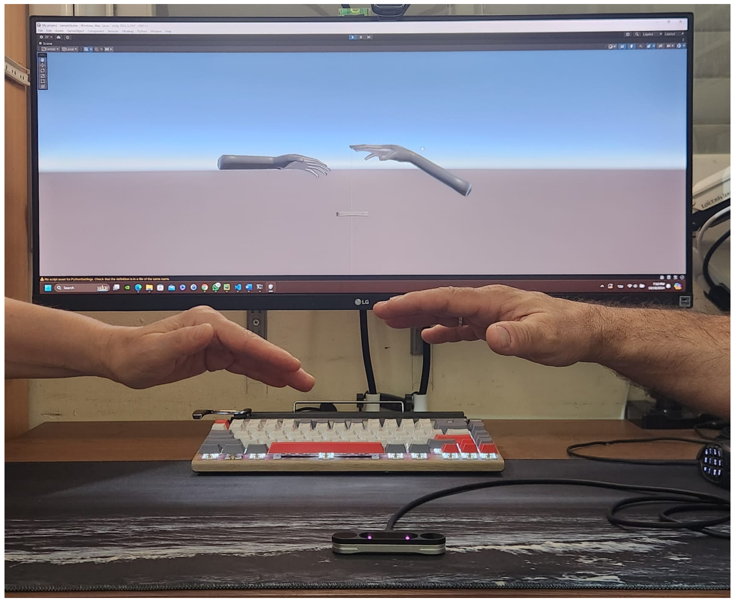
5.3 50/50 Synchronization

In this case, we want to see that even if we do not start syncing and match up at a certain point the algorithm will pick up on this. So we will start with random hand movements for the first half of the recording and then match up with a set movement for the second half.

5.4 The Mirror Game

The mirror game is a common mimicry exercise where two players mimic each other's full-body movements. It is often used in theater dance and movement therapy. It is one of the first game methodologies that was developed for research purposes to measure synchronization stats of togetherness. In the original mirror game experimental setting, two players faced each other holding handles that can move along parallel tracks. The players were instructed to move together in a synchronized and interesting manner. According to each player's velocity profile, the player's synchronization was calculated according to the mean relative difference in velocity and the timing difference between zero-velocity events [20,31]. In the following years, many studies extended this paradigm and measured movement using different technologies. In this case, the basis of the game will stay the same however we will only be looking at the hand movements of the participants.

The following image shows how hand recordings were created using the Leap Motion Controller and how Unity displays the hands. Two people face each other, extending their hands over the LMC.



6 Synchronization Metrics

Synchronization can be quantified using several objective mathematical measures, as the concept is often interpreted in different ways. One common method for evaluating synchronization is by analyzing the correlation between two-time series that represent the movements of individuals [29,32]. A time series can track various aspects of movement, such as velocity, with some studies using data from markers attached to participants' bodies or hands. Other approaches measure acceleration using accelerometers attached to the body or head. In tasks where participants are asked to tap in sync, time series data can be obtained by recording the intervals between taps, known as inter-tap intervals. A novel metric called dynamic pose similarity (DPS) has recently been introduced, aggregating the positions of 15 joints and their movement directions to generate a time series [30].

For studies using video recordings, individual body movements are often analyzed with Motion Energy Analysis (MEA), which tracks changes in pixel intensity between consecutive video frames. This method generates a time series of motion intensity values for each participant [23, 28]. A widely used technique in time series analysis is the Rolling-Window Time-Lagged Cross-Correlation (RWTLCC), which calculates correlations between two data streams at different time lags. This approach accounts for the possibility that the time series are not perfectly aligned and evaluates correlations across a range of time lags.

Other methods for measuring synchronization include Pearson correlations, dynamic time warping (DTW), and phase synchrony [33, 34]. Each of these methods captures different aspects of synchronization. For example, they may evaluate the overall strength of synchronization throughout an interaction or focus on the frequency of synchronized intervals.

In addition to quantifying synchronization, some approaches aim to classify synchronization states. This can be particularly useful in therapeutic settings, such as a study that assessed social engagement in autistic children by measuring interpersonal movement synchrony during a theatrical workshop. Wrist-worn accelerometers captured movement data, and synchronization was calculated using cross-wavelet similarity comparisons [28]. In classification tasks, machine learning techniques can provide valuable insights for identifying synchronized states.

7 Algorithms

7.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is an algorithm used to measure the similarity between two temporal sequences that may vary in time or speed. This makes it particularly useful for comparing sequences where one might be faster or slower than the other, such as hand movements recorded in different conditions. The key idea behind DTW is to non-linearly warp the time axis of one sequence to align it with the other, minimizing the total distance between corresponding points.

Mathematically, DTW calculates a cumulative distance matrix where each element is computed as:

$$D_{i,j} = d(x_i, y_j) + \min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$$

where $D_{i,j}$ is the accumulated distance at point (i, j), and $d(x_i, y_j)$ is the Euclidean distance between points x_i and y_j. This process ensures that the sequences are aligned in the best possible way, even if one sequence is temporally stretched or compressed.

In this equation, the first term $d(x_i, y_j)$ represents the local cost, typically the Euclidean distance between the current points x_i and y_j from the two sequences. The second part of the equation, $\min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$, determines the path that results in the minimum cumulative distance. The algorithm chooses the optimal alignment by exploring three possible transitions:

- $D_{i-1,j-1}$: Moving diagonally, which corresponds to matching the current points x_i and y_j from both sequences.
- $D_{i-1,j}$: Moving vertically, which corresponds to inserting a point in the second sequence y_j.
- $D_{i,j-1}$: Moving horizontally, which corresponds to inserting a point in the first sequence x_i.

This recursive process builds up the cumulative distance matrix D, where each entry $D_{i,j}$ represents the minimum cost of aligning the subsequences up to points x_i and y_j. The final DTW distance is found at $D_{n,m}$, where n and m are the lengths of the sequences, representing the overall alignment cost. This dynamic programming approach allows for flexible alignment, even when the sequences vary in length or timing.

In our project, DTW is applied to analyze the synchronization of hand movements captured using the Leap Motion controller. To capture the full scope of hand movements, we extracted several key features from the hand data: palm positions, palm velocities, fingertip positions, wrist positions, and individual finger bone positions. Each of these metrics was represented as a time series, allowing for direct comparison between the right and left hands using DTW.

One of the significant advantages of DTW is its ability to handle variations in timing. For example, if one hand moves slightly faster or slower than the other, DTW compensates for this by stretching or compressing the sequences in time, ensuring that the most similar movements are aligned. This is especially important in our analysis, where perfect synchronization between two hands is unlikely.

For each feature—palm position, velocity, fingertip position, wrist position, and finger bone position—DTW was used to calculate the distance between the sequences. We introduced a margin of error to account for small, natural variations in human movement. Additionally, we employed a sliding window approach, allowing for comparisons between frames that are close in time but not exactly aligned. This technique further improved the robustness of the analysis, ensuring that brief delays or slight offsets in movement were taken into account.

The results of this DTW analysis provide a normalized distance measure, which indicates how synchronized the two hands are. Lower DTW distances suggest higher synchronization, while higher distances reflect more significant discrepancies in movement patterns. This method allows us to quantify the degree of synchronization across multiple aspects of hand movement, providing a comprehensive view of how well the hands align in time and space.

7.2 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) is a statistical method used to explore the relationships between two sets of variables. Unlike simpler correlation techniques that assess the similarity between individual pairs of variables, CCA looks for linear combinations of the variables in both datasets that maximize their mutual correlation. This makes CCA particularly useful in multivariate analysis, where multiple variables from two datasets may be related in complex ways. In the context of our project, CCA helps us measure the synchronization between hand movements by comparing multiple aspects of the hand simultaneously, such as finger movements and the vectors from the palm to the fingertips.

Mathematically, CCA seeks to find two sets of linear transformations that maximize the correlation between two datasets $\mathbf{X}$ and $\mathbf{Y}$ by solving the following:

$$\rho = \max_{\mathbf{w}_x, \mathbf{w}_y} \frac{\mathbf{w}_x^\top \mathbf{X}\mathbf{Y}^\top \mathbf{w}_y}{\sqrt{\mathbf{w}_x^\top \mathbf{X}\mathbf{X}^\top \mathbf{w}_x}\sqrt{\mathbf{w}_y^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{w}_y}}$$

where $\mathbf{w}_x$ and $\mathbf{w}_y$ are the weight vectors for the two datasets that maximize the correlation ρ between the two linear combinations of the variables in $\mathbf{X}$ and $\mathbf{Y}$.

In this equation, $\mathbf{w}_x^\top \mathbf{X}\mathbf{Y}^\top \mathbf{w}_y$ represents the covariance between the two linear combinations of the variables in $\mathbf{X}$ and $\mathbf{Y}$, while the denominator contains square roots to normalize this covariance by the variances of each linear combination.

The terms $\sqrt{\mathbf{w}_x^\top \mathbf{X}\mathbf{X}^\top \mathbf{w}_x}$ and $\sqrt{\mathbf{w}_y^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{w}_y}$ represent the standard deviations of the two linear combinations. These terms ensure that the resulting value of ρ is a *correlation coefficient* that is bounded between -1 and 1.

The square root of the variance (which gives the standard deviation) plays a critical role in normalizing the linear combinations of $\mathbf{X}$ and $\mathbf{Y}$. Without this normalization, the equation would represent raw covariance, which could be affected by the scale or units of the original variables. By taking the square root of the variances, we ensure that ρ is independent of the scales of the variables and represents the *pure strength of the relationship* between the two sets of variables. In other words, the square root terms are necessary to transform the covariance into a dimensionless measure that can be interpreted as correlation.

In our study, we used CCA to assess the synchronization of hand movements captured by the Leap Motion Controller. Specifically, we focused on comparing two main sets of data: the bone vectors of each finger and the vectors formed from the palm to the fingertip. For each frame of data, these vectors represent the movements and orientations of the hand's skeletal structure in 3D space. By applying CCA, we were able to quantify the relationship between the two hands—one recorded directly, and the other mirrored along the x-axis to simulate a mirror-image scenario.

The first step involved extracting the bone vectors of the five fingers from each hand and comparing the corresponding fingers using CCA. Since CCA works best when the datasets are of equal length, we truncated the datasets to the shortest sequence to ensure a fair comparison. After preprocessing, the algorithm computed the correlation for each finger by finding the optimal linear transformations that align the movements of the two hands. The result was a correlation value for each finger, which reflects how synchronized the finger movements were between the two participants.

Additionally, we used CCA to compare the vectors formed between the palm and each fingertip, which represent another aspect of hand movement. These palm-to-tip vectors were also compared between the two hands after applying the mirror transformation to one of the hands. CCA was again applied to find the best-fitting linear transformations and to calculate the correlation between these vectors.

In both cases, the results provided insight into how well the movements of the two hands were synchronized. High correlation values indicated strong synchronization, while lower values reflected discrepancies between the hands. This approach, using CCA, allowed us to capture complex multivariate relationships in the hand movements, which are difficult to assess using simpler methods like basic correlation or distance metrics.

7.3 Cross-Correlation

Cross-correlation is a technique used to measure the similarity between two signals as a function of the time lag applied to one of them. In the context of hand movement synchronization, cross-correlation allows us to determine how well the movements of the left and right hands align, even if there is a slight delay between them. The basic idea is to shift one signal in time relative to the other and compute the similarity at each shift. The maximum value of this similarity (or correlation) provides insight into the level of synchronization between the two movements.

Mathematically, the normalized cross-correlation between two signals $a[n]$ and $b[n]$ with a lag τ is given by:

$$R_{ab}(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{(a[n] - \mu_a)(b[n+\tau] - \mu_b)}{\sigma_a \sigma_b}$$

where μ_a and μ_b are the means of a and b, and σ_a and σ_b are their standard deviations. This function measures the similarity at different time shifts (lags) and helps identify the time delay where the two signals are most synchronized.

In this equation, $R_{ab}(\tau)$ calculates the correlation between the signals $a[n]$ and $b[n+\tau]$ at each lag τ, while the means μ_a and μ_b center the signals, and the standard deviations σ_a and σ_b normalize the result. This normalization ensures that the cross-correlation values are comparable across different signals, with a maximum value of 1 representing perfect synchronization.

In this project, we applied cross-correlation to compare several aspects of hand movements, including palm positions, palm velocities, fingertip positions, wrist positions, and finger bone positions. For each of these features, we performed a normalized cross-correlation to determine the maximum correlation and the time lag at which it occurred. This method helped us identify whether the hands were moving in sync and if there was a delay between their movements.

The process begins by extracting the relevant features from the Leap Motion controller data. These features include the 3D coordinates for palm, wrist, fingertip, and finger bone positions, as well as velocities. Since we are comparing mirrored hand movements, the data for the left hand is flipped along the appropriate axis to simulate a real-world mirror reflection.

Next, cross-correlation is applied to each feature to compute the similarity between the left and right hands at various time shifts (lags). The algorithm computes the maximum correlation, which indicates the highest level of synchronization, and identifies the lag at which this occurs. If the maximum correlation occurs at a lag of zero, it suggests that the movements are synchronized in real time. A non-zero lag indicates that one hand is leading or lagging behind the other, but still following a similar movement pattern.

In addition to cross-correlation with lag, we also calculate the correlation without any time shift. This gives us a baseline for how well the hands are synchronized when considered simultaneously, without adjusting for potential delays. Comparing both lagged and non-lagged correlations allows us to better understand whether the movements are naturally in sync or if they require a temporal shift to align.

This approach provides a comprehensive view of hand movement synchronization across multiple features. The results of the cross-correlation analysis can be used to quantify the degree of synchronization and detect any time lags between the hands, offering insights into how well the movements of the two hands match in both time and space.

7.4 Wavelet Coherence

Wavelet Coherence is a method used to analyze the synchronization between two signals across both time and frequency domains. It provides a detailed understanding of how the similarity between two signals changes over time and at various frequencies, making it ideal for capturing the complex nature of hand movements. In our project, we used wavelet coherence to compare the palm positions and velocities of the left and right hands, as recorded by the Leap Motion controller.

Mathematically, the wavelet coherence between two signals $a(t)$ and $b(t)$ is calculated as:

$$W_{ab}(s,t) = \frac{|S(W_a(s,t)W_b^*(s,t))|^2}{S(|W_a(s,t)|^2)S(|W_b(s,t)|^2)}$$

where $W_a(s,t)$ and $W_b(s,t)$ are the wavelet transforms of $a(t)$ and $b(t)$, and S is a smoothing operator. This calculation provides insight into the similarity of the two signals at various time points and frequencies.

In this equation, $W_a(s,t)$ and $W_b(s,t)$ represent the wavelet transforms of signals $a(t)$ and $b(t)$, while S applies a smoothing operator to ensure meaningful coherence estimates. The numerator captures the cross-spectrum between the wavelet transforms of the two signals, and the denominator normalizes this by the individual power spectra. The result is a coherence value between 0 and 1, where 1 indicates perfect synchronization.

In this case, we applied the Complex Morlet Wavelet (more) for the wavelet transform, which is particularly useful for time-frequency analysis. This allows us to observe how hand movements are synchronized not only at different points in time but also across different movement frequencies. The left hand's data was flipped along the x-axis to simulate the mirror effect, ensuring that the comparison accurately reflected the real-world mirroring of hand movements.

To perform the analysis, we extracted three key features from the hand movement data: palm positions, velocities, and fingertip positions. For each feature, we compared the right and left hands along the three primary axes—X (left-right), Y (up-down), and Z (forward-backward). By analyzing these axes separately, we could gain insights into the specific dimensions in which the hands are synchronized.

The wavelet coherence was calculated for each axis, providing a measure of how synchronized the hand movements were in both time and frequency. Peaks in the coherence indicate strong synchronization at particular time points and frequency bands, while dips in the coherence suggest weaker alignment. This level of detail is valuable because it allows us to detect not only whether the hands are synchronized, but also at what points in time and across which frequencies the synchronization occurs.

The results of this analysis were visualized in coherence plots, with separate graphs for each axis (X, Y, Z). These plots give a clear picture of how well the hands are synchronized along each movement dimension, revealing both short-term and long-term synchronization patterns. This method provides a comprehensive view of hand movement synchronization, allowing us to understand not just if the hands are moving together, but how their movements align over time and across different frequencies.

However, the results obtained from this wavelet coherence analysis were not as accurate or meaningful as anticipated. After further investigation, it became clear why this method did not yield the expected output. The primary reason is related to the nature of the data itself. Wavelet coherence is designed to detect synchronization in signals that have cyclic or oscillatory behavior—that is, signals that exhibit clear frequency variations over time. However, the data we were analyzing consists of 3D spatial locations (x, y, z coordinates) representing the positions and velocities of the hands. These values change over time, but they do not exhibit the kind of periodic or oscillatory frequency patterns that wavelet analysis is suited for.

Hand movements, as represented by these 3D coordinates, are more translational or sporadic than cyclic. This means that the movements lack the repeated frequency components that wavelet coherence relies on to measure synchronization. In our case, the changes in hand position and velocity are not oscillatory but instead reflect changes in location, which makes them ill-suited for frequency-domain analysis. Consequently, the wavelet coherence algorithm could not detect any significant patterns of synchronization, resulting in coherence values close to zero or out of range.

In summary, while wavelet coherence is an effective tool for analyzing frequency-based synchronization in signals, it is not appropriate for spatial data such as the hand movements recorded in this project. The lack of oscillatory behavior in the 3D hand movement data explains why the wavelet algorithm did not produce the desired outputs.

8 Results

As described in the "Experiment Recordings" section, we used four different types of hand recordings, each lasting 15 seconds. Each type of recording was performed twice, resulting in a total of eight recordings. All the raw data is available at the end of the article in the appended tables. Out of the four algorithms mentioned in this paper, three proved successful, each with its own criteria. The average of these criteria will represent the algorithm's final result.

No single algorithm can accurately represent full synchronization or the lack thereof, as each algorithm captures a different aspect of synchronization:

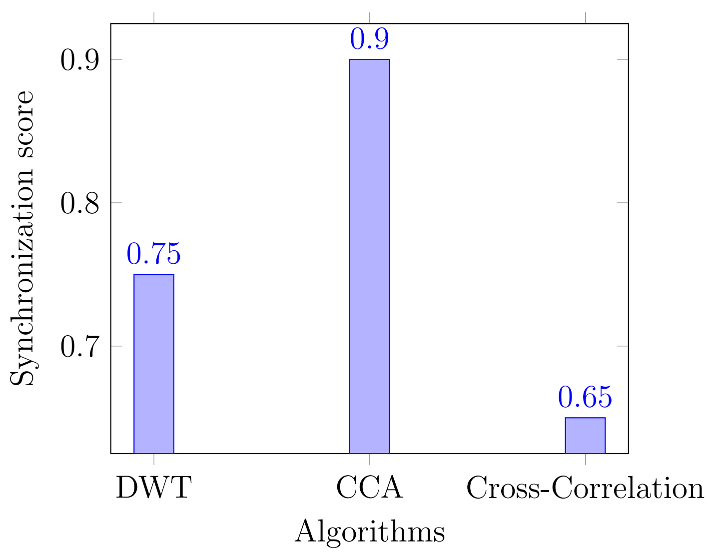
- DWT represents the general and overall synchronization of hand location relative to the Leap Motion Controller (LMC).
- CCA reflects the grip posture and direction of the hand, indicating how open or closed, flexed, or clenched the hand is, and the direction it is facing.
- Cross-Correlation measures the timing and alignment of hand movements, helping detect delays or shifts in movement patterns between

the two hands. It captures how synchronized the movements are over time and can highlight subtle misalignments in motion.

For this reason, each algorithm will represent one-third of the total score for each recording. The following tables show the score given to each algorithm on a scale from 0 to 1. Each score represents the synchronization percentage according to each algorithm.

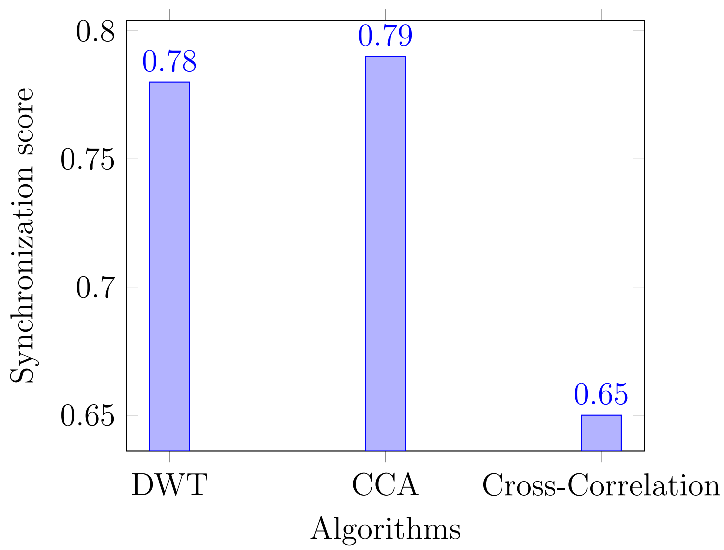
8.1 Mirror Game Recording 1

Reviewing The Mirror Game (see Section 4.4), given these numbers, we can conclude that the person assigned to be the follower managed to achieve relatively good spatial location, as indicated by the DWT score, and a good hand grip posture and direction, as reflected by the CCA. However, there is a slight lag in timing, as revealed by the Cross-Correlation. Giving us a total score of approximately 76% synchronization.



8.2 Mirror Game Recording 2

Reviewing the second recording of the Mirror Game experiment, we can see that the numbers do not change much. The results again show a relatively good spatial location, as indicated by the DWT score, and a very good hand grip posture and direction, as reflected by the CCA. However, there is once again a slight lag discrepancy. With a total score of 74% synchronization.

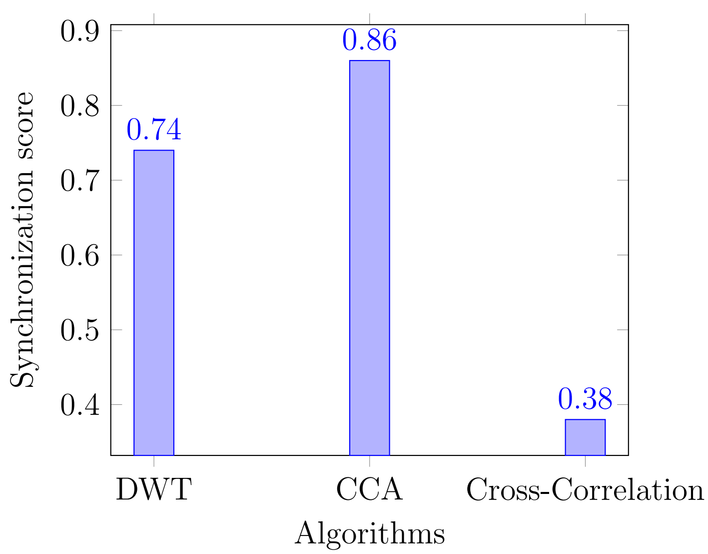


Finally, reviewing both recordings of the Mirror Game experiment, the data shows no issues in mimicking the hand grip posture and direction, a relatively good score when mimicking the overall hand location. However, based on the Cross-Correlation results, we can conclude that there is a significant delay in doing so, likely due to the person's reaction time to what they see. An improvement to consider would be increasing the duration of the experiment, extending the recording time from 15 seconds to 30 seconds or longer, which may help reduce this delay, by giving the participants more time to sync.

8.3 Half Unsynced Half synced Recording 1

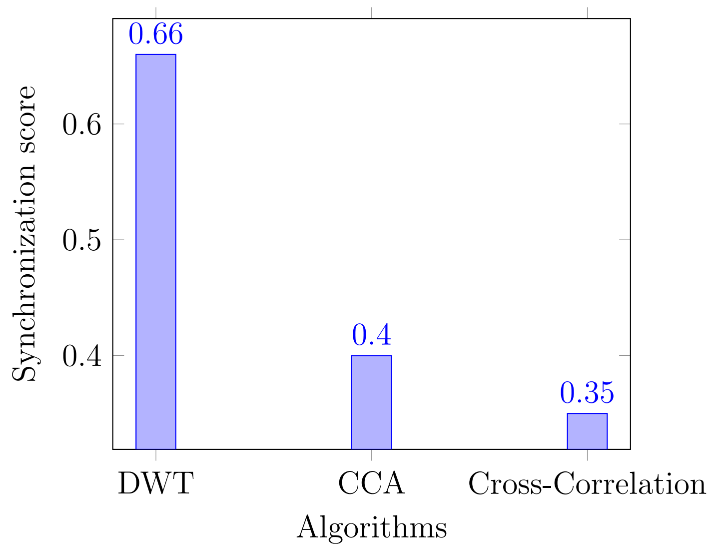
Looking at the following chart, we are reviewing the Half Unsynced Half Synced recording experiments (Section 4.3). Since we are combining synchronized and unsynchronized movements, we didn't really know what to expect. It is important to note that the unsynchronized movements in the first recording were planned, meaning they were unsynchronized but not random, for several reasons. The primary reason was to ensure that the initial movements would be fully unsynchronized, making it easier to sync in the middle of the recording. However, in the second recording, the unsynchronized movements were not planned and were entirely random.

Looking at the numbers for the first recording, we see that the overall location of the hands, as shown by the DWT, is actually quite good. This was likely due to the planned unsynchronized movements, as there were no spontaneous or aggressive actions. Surprisingly, the same was true for the CCA algorithm. However, when examining the Cross-Correlation scores, we observe a significant decrease, as expected, suggesting there is a substantial time lag, as anticipated. This gives us a final synchronization score of 66% for this experiment.



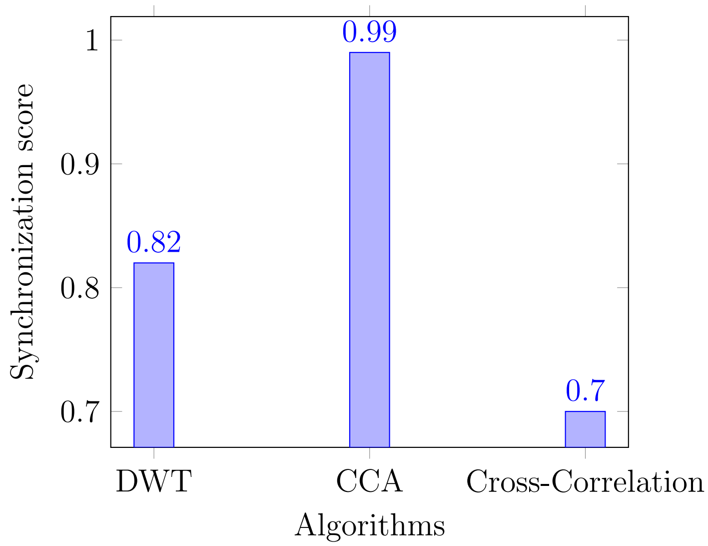
8.4 Half Unsynced Half synced Recording 2

Looking at the second recording of the Half Unsynced Half Synced experiment, we see that our suspicions were correct. While the DWT still shows slight spatial synchronization, the CCA and Cross-Correlation scores show a significant decrease in the numbers, reflecting almost no synchronization, with a massive time lag in the minimal synchronization that does exist. This is likely caused by the random movements of the hands before synchronization, making it harder for the algorithms to detect any synchronization that did occur. The final score for this recording is 47%.



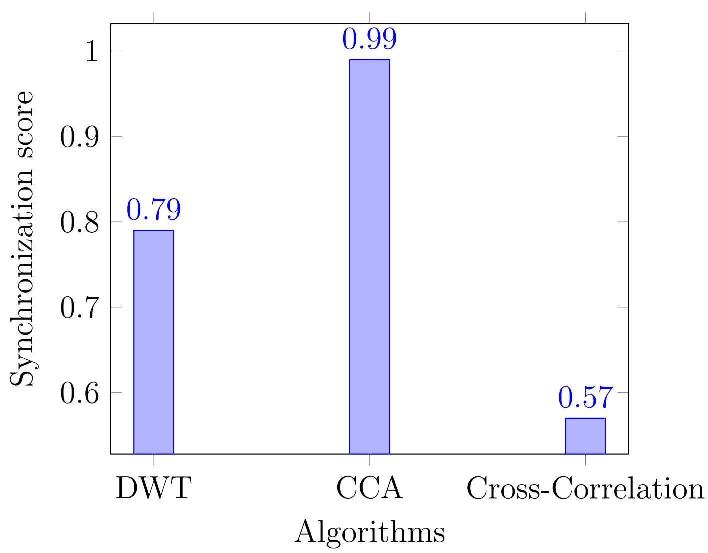
8.5 Synced Recording 1

In the following recordings, we expect to get the best results, as these will serve as a baseline for our experiments. As we can see across the board, all three algorithms show high results, indicating good overall hand location as shown by the DWT. The hand direction and grip posture are as good as they can be shown by the CCA. Nevertheless, there is still a slight lag, as expected shown by the Cross-Correlation. There will always be some discrepancies, as two hands can never move exactly like robots, in perfect synchronization both in time and space. The final score given to this recording is 83% synchronization.



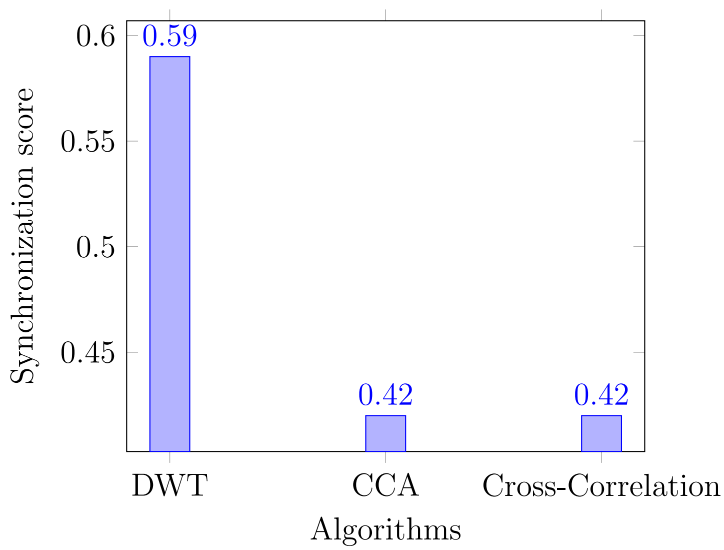
8.6 Synced Recording 2

As with the first recording of this experiment, we expected to see good results here. However, despite the DWT and CCA showing good results, the Cross-Correlation indicates that there is slightly more time lag than expected. Nevertheless, the final score for this recording is still high, at 78%.



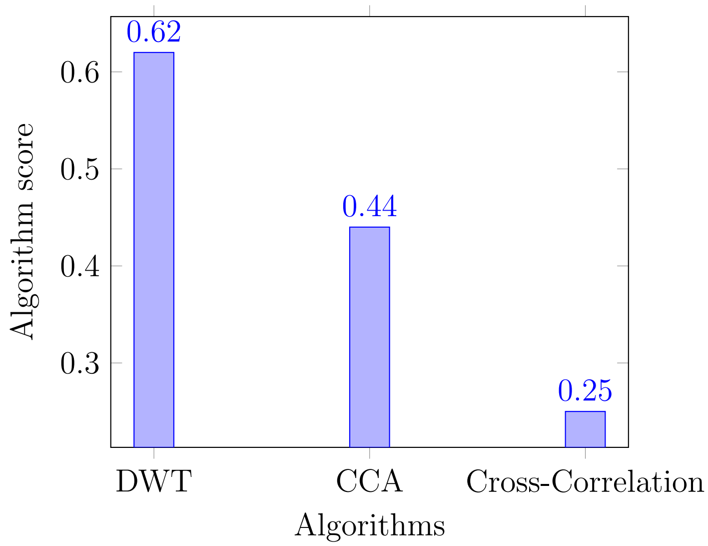
8.7 UnSynced Recording 1

When reviewing the following results, we were surprised by the outcome. We expected to see much lower numbers than what we actually received, given that the movements in this recording were chaotic and completely random. The DWT shows that the hand locations are still quite similar, although this may be due to the limited tracking space of the device itself. The CCA and Cross-Correlation, however, provide results more in line with our expectations, with the CCA indicating poor hand direction and grip posture. Even with time manipulation, there is no chance of achieving synchronization, as shown by the Cross-Correlation. This gives us a total synchronization score of 47%.



8.8 UnSynced Recording 2

Looking at the second recording, we can see similar results, confirming our initial analysis and once again providing us with a baseline for the worst possible outcome, with a final score of 43% synchronization. It is interesting to note that even with absolute randomness, there is still some degree of synchronization present.



Based on these results, we can conclude that none of these algorithms alone can accurately predict synchronization. Only by considering all three, and potentially exploring additional, unresearched algorithms, can we effectively analyze hand movement synchronization. Furthermore, within each algorithm, when extracting the features representing hand movement, each feature was given equal weight in the final score. This may not be the most efficient approach, as different features may have more significance than others. Nevertheless, after reviewing all the recordings and experiments, we conclude that synchronization can be accurately measured using the Leap Motion device.

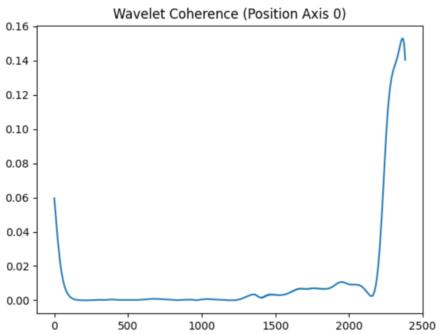
9 Data tables

9.1 Mirror Game Recording 1

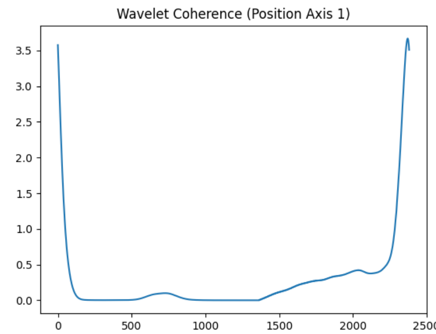
DWT	Total Distance	Normalized Distance
Palm position	716.1940529972884	0.30066920780742584
Wrist Position	984.2610370717297	0.41320782412751034
Velocity	369.7792867638128	0.1552389952828769
Tip Position	351.5778473087105	0.1475977528584007
THUMB (Finger 1)	654.4646881560145	0.2747542771435829
INDEX (Finger 2)	489.8772309499578	0.20565794750208136
MIDDLE (Finger 3)	516.9239202167997	0.21701256096423163
RING (Finger 4)	592.3183518449212	0.24866429548485355
PINKY (Finger 5)	710.0161232266022	0.29807561848304037

	CCA
THUMB	0.8192547012127728
INDEX	0.9158424873386126
MIDDLE	0.9165833298399183
RING	0.9069987563341959
PINKY	0.9328352599429036
Palm-to-Tip THUMB	0.8637719585447696
Palm-to-Tip INDEX	0.9064117592350365
Palm-to-Tip MIDDLE	0.9167169866889104
Palm-to-Tip RING	0.8943826787575931
Palm-to-Tip PINKY	0.9411266921065042

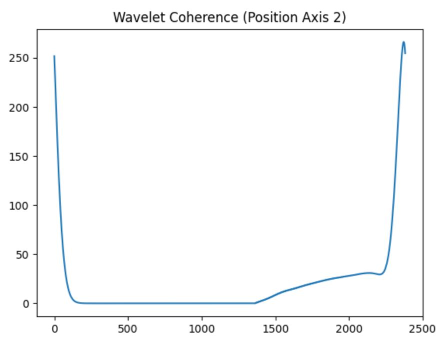
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.5578330847349237	144	0.5397521037854756
Wrist Position	0.5455089368075436	143	0.5286973361588039
Velocity	0.3999619167574598	993	0.1444360316504262
Tip Position	0.688021177333952	153	0.6530403302633194
THUMB (Finger 1)	0.778447021763568	148	0.7518160372799192
INDEX (Finger 2)	0.7353704548530071	148	0.7076992817404856
MIDDLE (Finger 3)	0.7025793186011146	151	0.6719649969239049
RING (Finger 4)	0.7444009349015857	149	0.7157273277616033
PINKY (Finger 5)	0.7460264371001809	149	0.7160944380383633



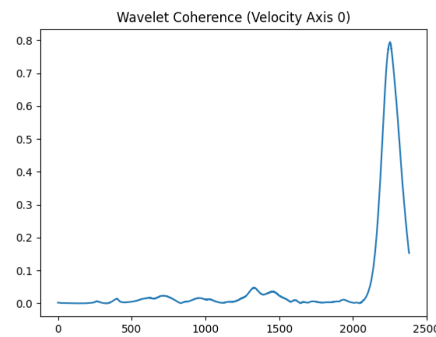
(a) Position Axis X



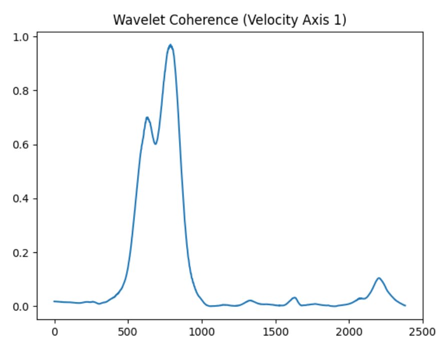
(b) Position Axis Y



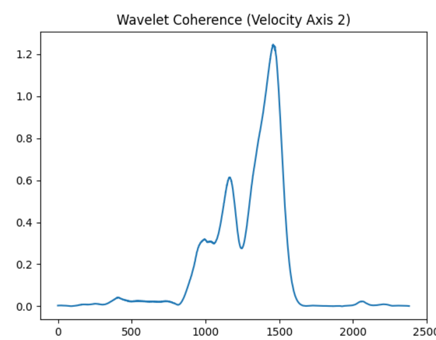
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

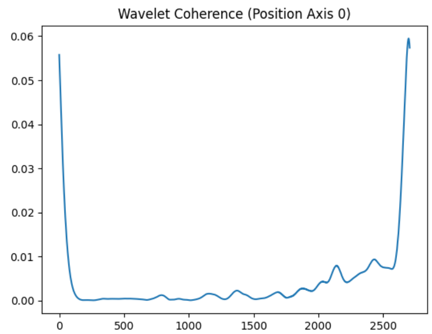
Figure 2: Wavelet Game 1

9.2 Mirror Game Recording 2

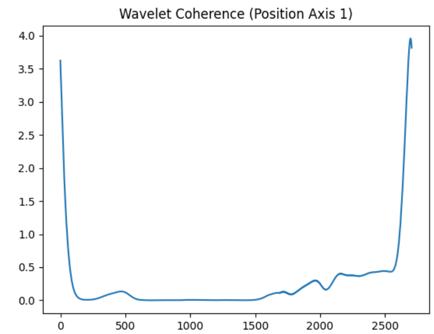
DWT	Total Distance	Normalized Distance
Palm position	701.5619898615348	0.25926163705156496
Wrist Position	994.8280735211663	0.3676378690026483
Velocity	566.1762557687464	0.2092299540904458
Tip Position	303.1525480573282	0.11202976646612277
THUMB (Finger 1)	625.7319298432802	0.2312387028245677
INDEX (Finger 2)	453.3594767628033	0.16753860929889258
MIDDLE (Finger 3)	481.0120422407564	0.17775759136761138
RING (Finger 4)	559.6288709700601	0.20681037360312643
PINKY (Finger 5)	672.3837409934006	0.24847883998277923

	CCA
THUMB	0.6871659199369289
INDEX	0.8418028103846595
MIDDLE	0.8077475392584681
RING	0.7890422298520661
PINKY	0.7909208836767256
Palm-to-Tip THUMB	0.8346643809922453
Palm-to-Tip INDEX	0.8367464696554822
Palm-to-Tip MIDDLE	0.8201556676762293
Palm-to-Tip RING	0.7855109335313762
Palm-to-Tip PINKY	0.8026595342609597

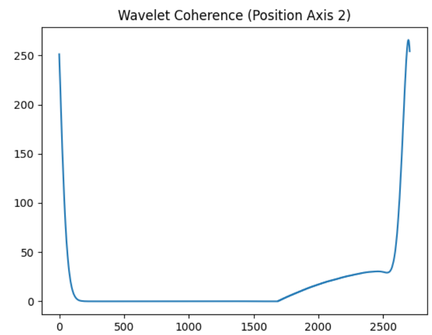
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.6068905822657097	105	0.543565096177731
Wrist Position	0.49725283750883426	108	0.4428312416677632
Velocity	0.29496590064861294	95	0.20277530333011703
Tip Position	0.740232360047195	67	0.7375029362681128
THUMB (Finger 1)	0.7451445793899266	61	0.7404388826730092
INDEX (Finger 2)	0.7454092938185405	65	0.7420718047860068
MIDDLE (Finger 3)	0.732460152851466	68	0.7335545933043659
RING (Finger 4)	0.7630009832815573	70	0.7628300888276978
PINKY (Finger 5)	0.7604813420478174	67	0.7571336635006086



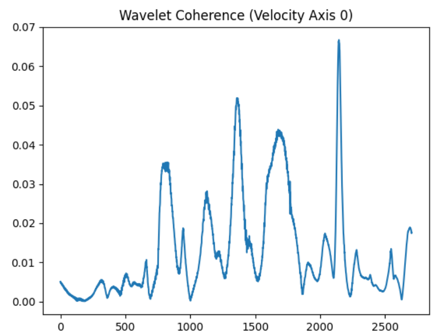
(a) Position Axis X



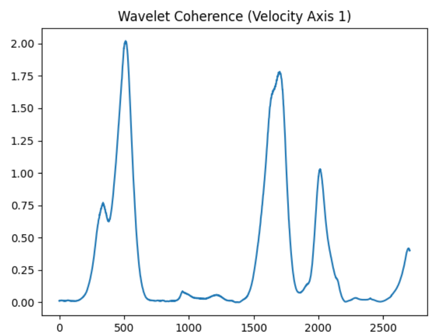
(b) Position Axis Y



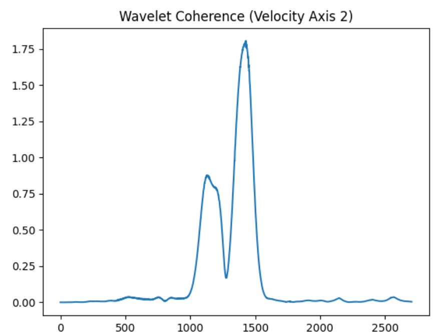
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

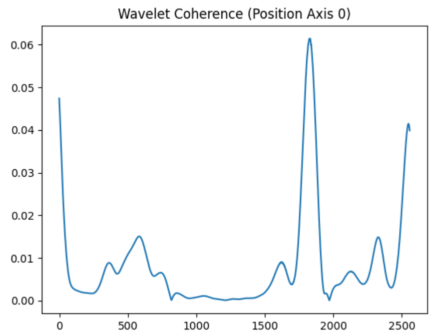
Figure 3: Wavelet Game 1

9.3 Half Unsynced Half synced recording 1

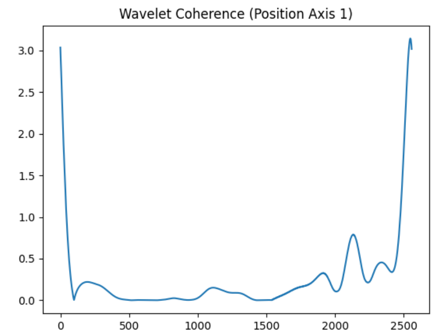
DWT	Total Distance	Normalized Distance
Palm position	653.2491859307399	0.25527518012143025
Wrist Position	877.3367148580006	0.34284357751387284
Velocity	1069.9113996303468	0.4180974598008389
Tip Position	501.17449999918483	0.19584779210597297
THUMB (Finger 1)	639.3037790675986	0.24982562683376264
INDEX (Finger 2)	516.2876988332079	0.20175369239281277
MIDDLE (Finger 3)	546.8128864805891	0.21368225341171906
RING (Finger 4)	596.7641485756055	0.2332020901037927
PINKY (Finger 5)	677.4182626321028	0.2647199150574845

	CCA
THUMB	0.814657182480102
INDEX	0.8774062994223993
MIDDLE	0.8730923243789448
RING	0.8737025784022134
PINKY	0.8703296191128106
Palm-to-Tip THUMB	0.8473020897877187
Palm-to-Tip INDEX	0.8979036560043335
Palm-to-Tip MIDDLE	0.8864532727866413
Palm-to-Tip RING	0.8780713593243569
Palm-to-Tip PINKY	0.8550246929705817

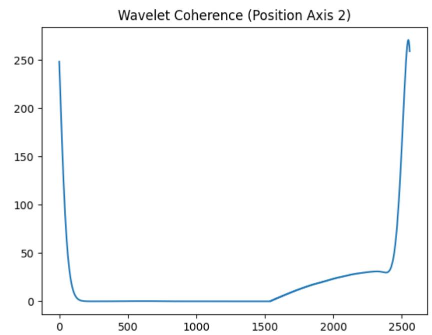
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.3287609844619893	1260	-0.16216413784604097
Wrist Position	0.403989725944037	1262	-0.1977386969814006
Velocity	0.43654749862288905	1064	-0.030961646044937095
Tip Position	0.43019463320143375	357	0.04283151817182539
THUMB (Finger 1)	0.33099075596740324	1022	-0.11128241705169188
INDEX (Finger 2)	0.3150370549502296	349	-0.06644674425404556
MIDDLE (Finger 3)	0.3632753580669202	349	-0.004897532723537727
RING (Finger 4)	0.4082722154540406	350	0.04558234071509982
PINKY (Finger 5)	0.46643609379929046	353	0.1575894794752275



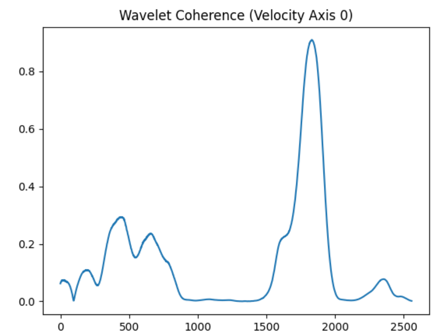
(a) Position Axis X



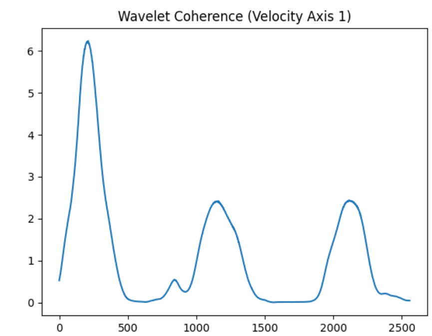
(b) Position Axis Y



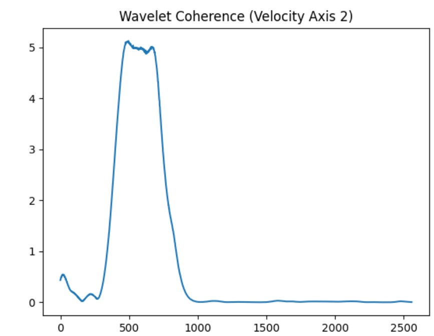
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

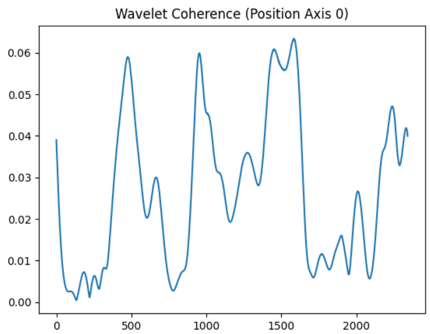
Figure 4: Wavelet 50/50 sync 1

9.4 Half Unsynced Half synced recording 2

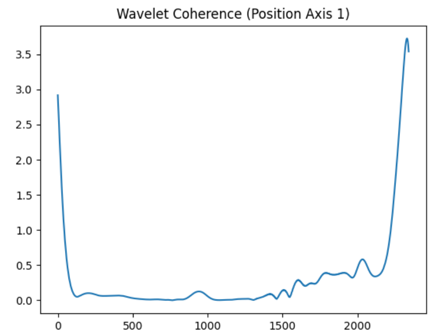
DWT	Total Distance	Normalized Distance
Palm position	677.9651486472337	0.289357724561346
Wrist Position	796.8639799144929	0.3401041314189044
Velocity	1960.7470842835094	0.8368532156566408
Tip Position	607.4875449578844	0.2592776546982008
THUMB (Finger 1)	660.4359622401414	0.2818762109432955
INDEX (Finger 2)	619.532610329928	0.2644185276696236
MIDDLE (Finger 3)	635.0251363761431	0.27103078803932695
RING (Finger 4)	654.2025864225523	0.2792157859251183
PINKY (Finger 5)	687.8116811660808	0.2935602565796333

	CCA
THUMB	0.21189308885740898
INDEX	0.2764906018164592
MIDDLE	0.38354019986851645
RING	0.46011169565473276
PINKY	0.49297516648102724
Palm-to-Tip THUMB	0.42131722194573834
Palm-to-Tip INDEX	0.2993902862580938
Palm-to-Tip MIDDLE	0.3871474533062668
Palm-to-Tip RING	0.5074886170842378
Palm-to-Tip PINKY	0.5598036227486193

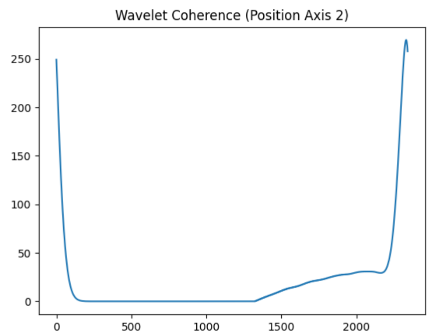
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.4422045632134396	-237	-0.20524645969124822
Wrist Position	0.3245675702770288	-242	-0.23620759451551007
Velocity	0.4164180550790087	-236	-0.24448837741539226
Tip Position	0.3358713502651596	-244	-0.25191749142380215
THUMB (Finger 1)	0.4104250232623948	-254	-0.2359462127206662
INDEX (Finger 2)	0.31017131739175685	-255	-0.2605562820849966
MIDDLE (Finger 3)	0.29209917776866456	780	-0.23513951285239768
RING (Finger 4)	0.32813936346681083	779	-0.24030264894779174
PINKY (Finger 5)	0.3568905407520549	779	-0.22966571326199842



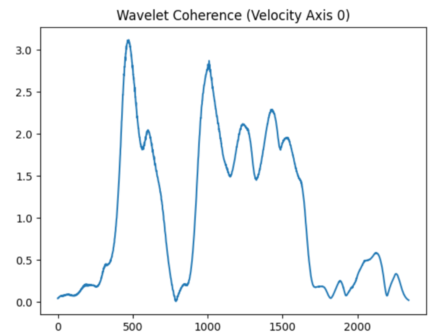
(a) Position Axis X



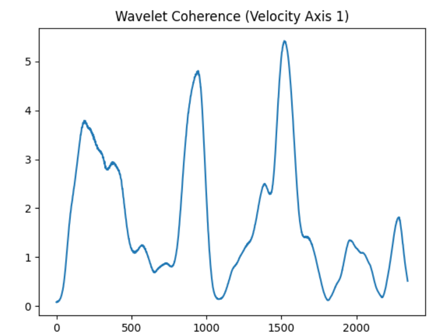
(b) Position Axis Y



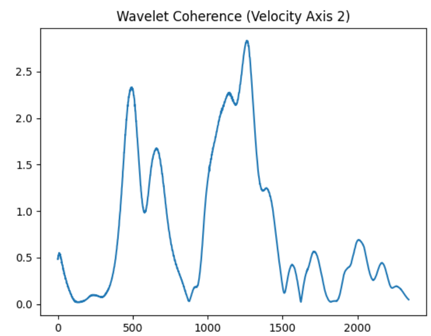
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

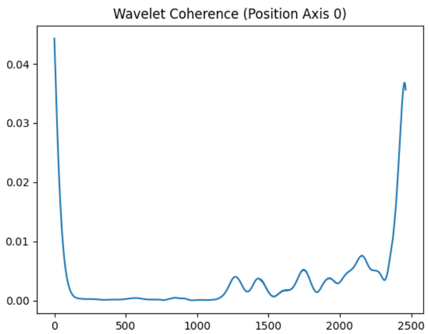
Figure 5: Wavelet 50/50 sync 2

9.5 Synced 1

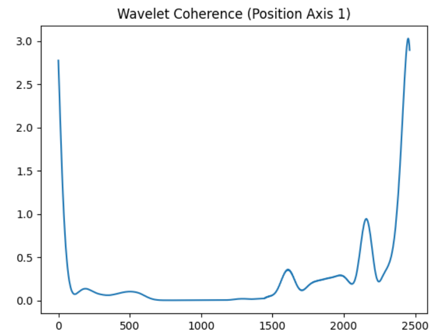
DWT	Total Distance	Normalized Distance
Palm position	541.3362923887057	0.2198766419125531
Wrist Position	813.1063915108892	0.33026254732367555
Velocity	203.38344087473328	0.08260903366154886
Tip Position	308.2432693314842	0.12520035309970926
THUMB (Finger 1)	558.0639338899562	0.22667097233548178
INDEX (Finger 2)	360.3065893494472	0.14634711183974297
MIDDLE (Finger 3)	379.591186900538	0.15418001092629488
RING (Finger 4)	432.8866048334043	0.17582721561064352
PINKY (Finger 5)	530.9335319750575	0.21565131274372765

	CCA
THUMB	0.9922397410473466
INDEX	0.9907344394292433
MIDDLE	0.9939187870125515
RING	0.995903053175966
PINKY	0.9916095499761047
Palm-to-Tip THUMB	0.9922243246474002
Palm-to-Tip INDEX	0.9880141915052707
Palm-to-Tip MIDDLE	0.9935031253872938
Palm-to-Tip RING	0.9956852510845062
Palm-to-Tip PINKY	0.9872127825080746

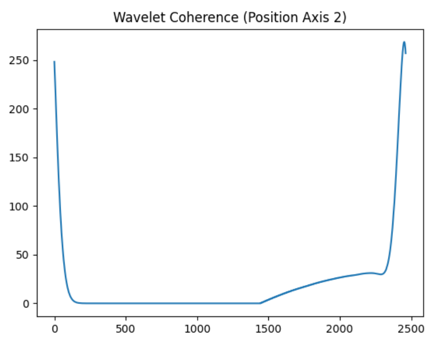
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.5535846037458416	704	0.3531220429928334
Wrist Position	0.5940417166585964	33	0.5790394330936665
Velocity	0.39433160906068926	-2	0.38476473602936906
Tip Position	0.7280848738603672	-1	0.727911359100083
THUMB (Finger 1)	0.49036321368966795	0	0.49036321368966784
INDEX (Finger 2)	0.767879843189124	0	0.7678798431891237
MIDDLE (Finger 3)	0.8505223491432604	0	0.8505223491432601
RING	0.9202552686664809	0	0.9202552686664809
PINKY	0.9634922838263201	0	0.9634922838263196



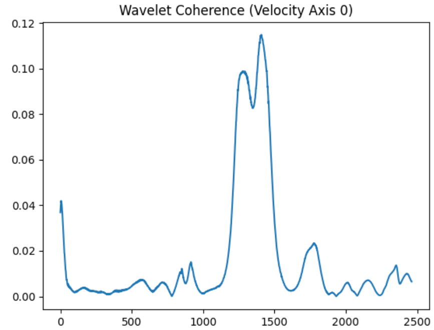
(a) Position Axis X



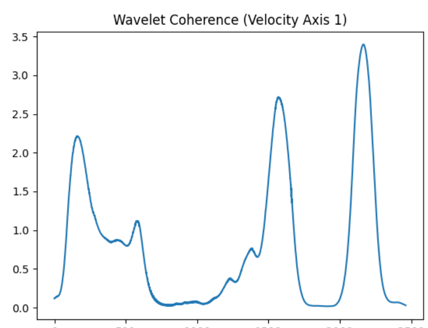
(b) Position Axis Y



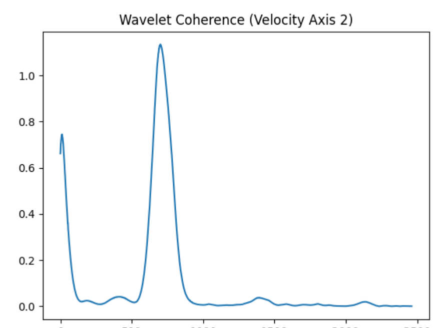
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

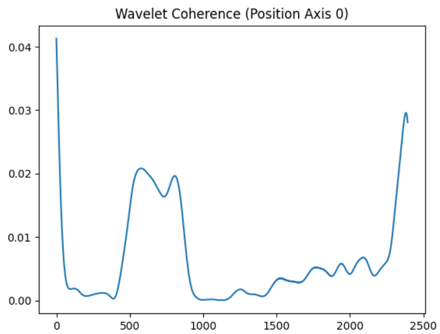
Figure 6: Wavelet Synced 1

9.6 Synced 2

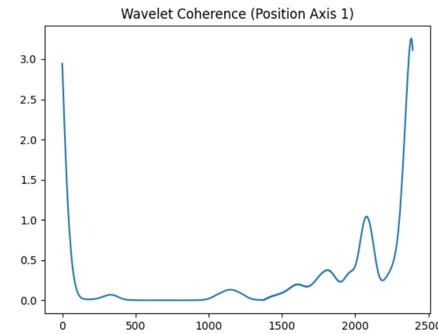
DWT	Total Distance	Normalized Distance
Palm position	575.6484892140182	0.24045467385715047
Wrist Position	824.5106560922214	0.3444071245163832
Velocity	253.68117790127826	0.10596540430295667
Tip Position	387.7982538764654	0.16198757471865724
THUMB (Finger 1)	560.8886863806855	0.23428934268199061
INDEX (Finger 2)	414.92319247633446	0.17331795842787573
MIDDLE (Finger 3)	447.8374499737882	0.1870666039990761
RING (Finger 4)	504.3096466740756	0.21065565859401655
PINKY (Finger 5)	594.7504755272898	0.2484337825928529

	CCA
THUMB	0.9874476652451321
INDEX	0.992614726804252
MIDDLE	0.99367975709218
RING	0.9944725071166448
PINKY	0.9929070635830923
Palm-to-Tip THUMB	0.9907405962380571
Palm-to-Tip INDEX	0.987688674655494
Palm-to-Tip MIDDLE	0.9933401364281061
Palm-to-Tip RING	0.9930697165516293
Palm-to-Tip PINKY	0.9914427126251142

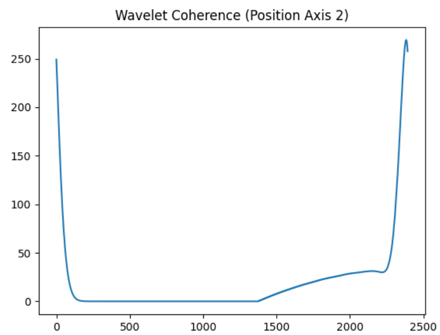
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.46418335693798757	817	-0.2499073614109882
Wrist Position	0.4743604150493671	584	-0.18256941780817068
Velocity	0.21899418056049486	647	0.07540684992227462
Tip Position	0.49237516285532934	-4	0.489556931009275
THUMB (Finger 1)	0.5363122409820634	-2	0.535254795854598
INDEX (Finger 2)	0.7535281313008797	0	0.7526910818993603
MIDDLE (Finger 3)	0.7553507592979749	0	0.7547580191971561
RING	0.7673522672604874	0	0.7666270641717146
PINKY	0.738290668508471	0	0.7378136168650822



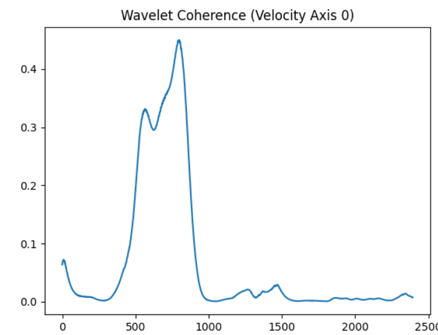
(a) Position Axis X



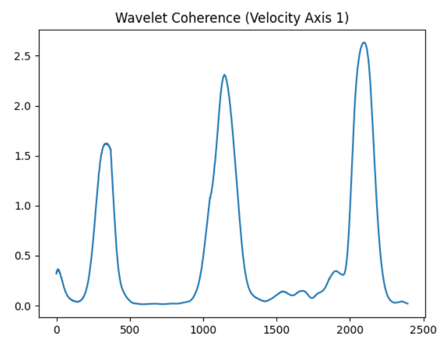
(b) Position Axis Y



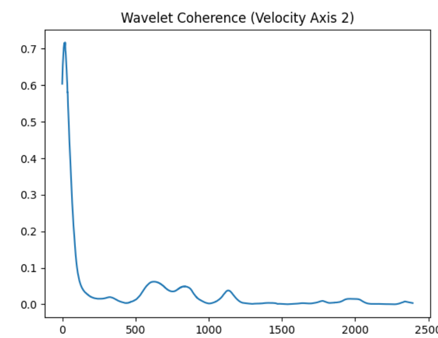
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

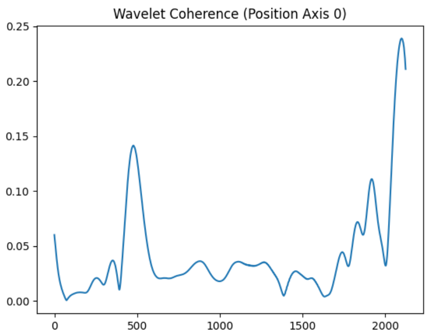
Figure 7: Wavelet 50/50 Synced 2

9.7 Unsynced 1

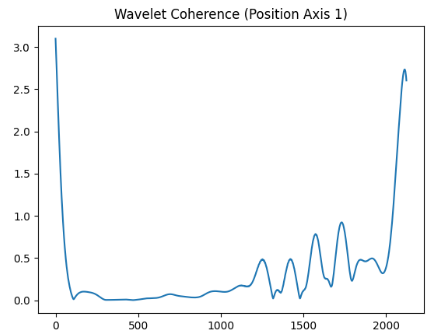
DWT	Total Distance	Normalized Distance
Palm position	751.0425031182721	0.3534317661733045
Wrist Position	857.9597546527206	0.40374576689539793
Velocity	2112.186987375046	0.9939703470000217
Tip Position	658.0997483559188	0.30969399922631474
THUMB (Finger 1)	728.2708341364765	0.34271568665245955
INDEX (Finger 2)	691.2789228695186	0.32530772840918526
MIDDLE (Finger 3)	700.1864200236896	0.3294994917758539
RING (Finger 4)	713.906652197514	0.3359560716223595
PINKY (Finger 5)	742.4226654818638	0.3493753719914653

	CCA
THUMB	0.3480823000414886
INDEX	0.4762582359582945
MIDDLE	0.4478618174857714
RING	0.42483741568356526
PINKY	0.40760646098384895
Palm-to-Tip THUMB	0.4060886341037927
Palm-to-Tip INDEX	0.47388018298800555
Palm-to-Tip MIDDLE	0.3763033412930445
Palm-to-Tip RING	0.3318503849507435
Palm-to-Tip PINKY	0.5077988801991214

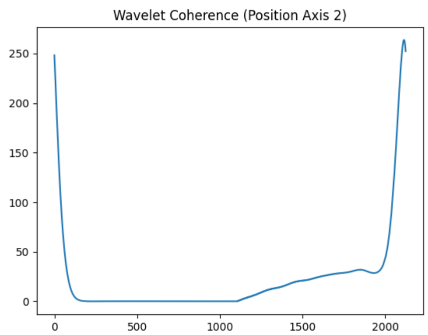
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.41811291682942775	455	-0.21579116431711642
Wrist Position	0.34534185907227666	454	-0.19602010704528477
Velocity	0.5580573195113953	66	-0.3199192191846587
Tip Position	0.36612595756459027	-475	-0.2842406472121903
THUMB (Finger 1)	0.5358252316746747	-114	-0.22222177286883601
INDEX (Finger 2)	0.39274582025229204	-112	-0.25802263475774356
MIDDLE (Finger 3)	0.36604551024213805	-113	-0.22156639819278107
RING	0.3931962321791109	-112	-0.18298693401375782
PINKY	0.45258446576596023	-110	-0.13714563163369828



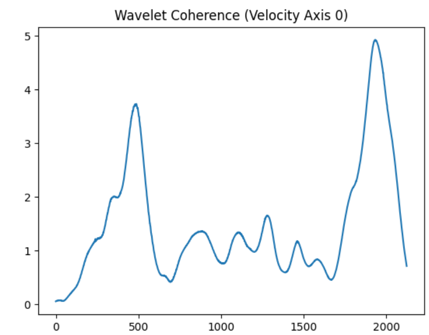
(a) Position Axis X



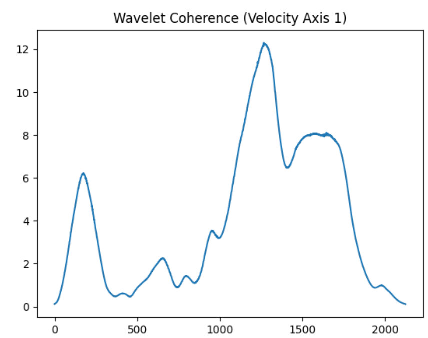
(b) Position Axis Y



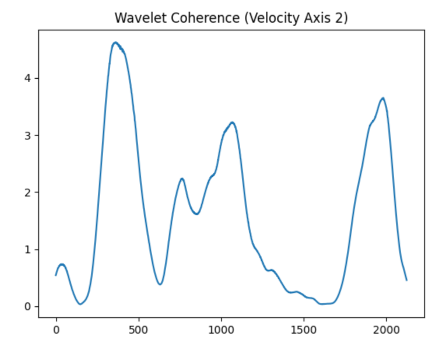
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

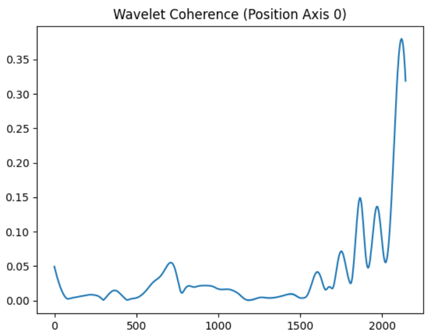
Figure 8: Wavelet unsunced 1

9.8 Unsynced 2

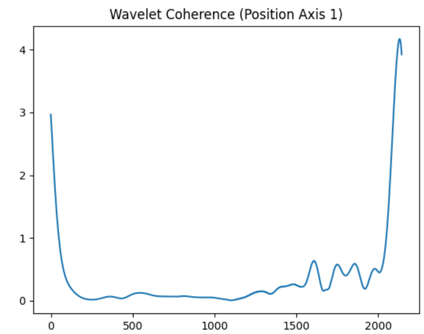
DWT	Total Distance	Normalized Distance
Palm position	681.2414287331251	0.31774320370015163
Wrist Position	795.7571817790517	0.37115540194918456
Velocity	2067.5927040022384	0.9643622686577604
Tip Position	578.5619626736983	0.26985166169482194
THUMB (Finger 1)	653.6581004384102	0.3048778453537361
INDEX (Finger 2)	621.0336768882881	0.28966122989192544
MIDDLE (Finger 3)	631.3900137626748	0.29449161089676995
RING (Finger 4)	645.3411187935302	0.3009986561536988
PINKY (Finger 5)	673.9976385338784	0.31436457021169706

	CCA
THUMB	0.39347236437599165
INDEX	0.4273198710990609
MIDDLE	0.4276594288324318
RING	0.4281388430487298
PINKY	0.46742404951500677
Palm-to-Tip THUMB	0.43382915783121034
Palm-to-Tip INDEX	0.42469092361770006
Palm-to-Tip MIDDLE	0.4547740947046779
Palm-to-Tip RING	0.48957620326340157
Palm-to-Tip PINKY	0.5195758338979517

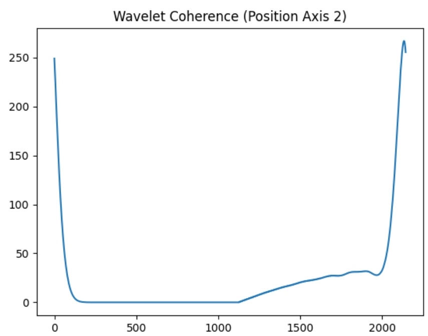
Cross-Correlation	Max Correlation with Lag	Lag	Correlation Without Lag
Palm position	0.31052626025888364	-137	0.11404782146662512
Wrist Position	0.29744408561075386	-462	0.21574656458795163
Velocity	0.331488920657386	106	0.13706307101138027
Tip Position	0.2255577607698157	592	-0.22165562502723757
THUMB (Finger 1)	0.2515191671425388	1147	-0.19012136107256547
INDEX (Finger 2)	0.22148441824368223	1794	-0.23448834511136077
MIDDLE (Finger 3)	0.20698352430285813	1796	-0.17293213595768311
RING	0.21501539844422757	-996	-0.14296440004300978
PINKY	0.2189862060695298	-995	-0.11458609422693398



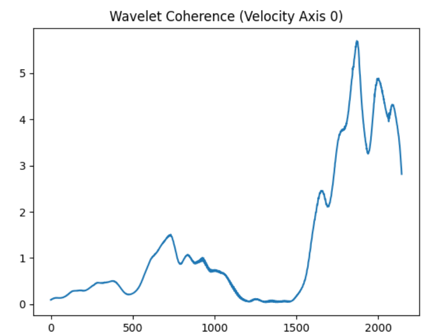
(a) Position Axis X



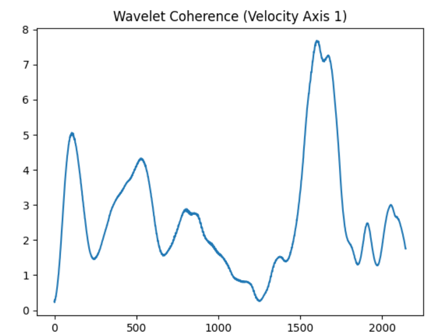
(b) Position Axis Y



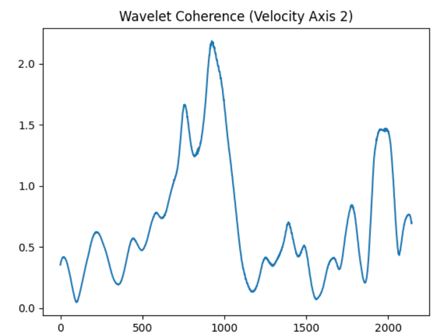
(c) Position Axis Z



(d) Velocity Axis X



(e) Velocity Axis Y



(f) Velocity Axis Z

Figure 9: Wavelet unsunced 2

References

[1] A. B. Johnston and D. C. Burnett. *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012.

[2] F. Zhang et al. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[3] F. Weichert et al. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[4] J. Guna et al. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors*, 14(2):3702–3720, 2014.

[5] G. Marin et al. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569, 2014.

[6] V. Bazarevsky et al. Blazeface: Sub-millisecond neural face detection on mobile gpus. *arXiv preprint arXiv:1907.05047*, 2019.

[7] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021.

[8] F. Ramseyer and W. Tschacher. Nonverbal synchrony in psychotherapy: Coordinated body movement reflects relationship quality and outcome. *Journal of Consulting and Clinical Psychology*, 79(3):284, 2011.

[9] S. R. Fussell and L. D. Setlock. Computer-mediated communication. In *The Oxford Handbook of Language and Social Psychology*, pages 471–489. 2014.

[10] S. H. Strogatz. *Sync: The emerging science of spontaneous order*. Hyperion Books, 2003.

[11] F. J. Bernieri and R. Rosenthal. Interpersonal coordination: Behavior matching and interactional synchrony. In R. S. Feldman and B. Rimé, editors, *Fundamentals of nonverbal behavior*, pages 401–432. Cambridge University Press, 1991.

[12] M. J. Hove and J. L. Risen. It's all in the timing: Interpersonal synchrony increases affiliation. *Social Cognition*, 27(6):949–960, 2009.

[13] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192, 2004.

[14] R. Feldman. Parent-infant synchrony and the construction of shared timing; physiological precursors, developmental outcomes, and risk conditions. *Journal of Child Psychology and Psychiatry*, 48(3-4):329–354, 2007.

[15] J. N. Bailenson. Nonverbal overload: A theoretical argument for the causes of zoom fatigue. *Technology, Mind, and Behavior*, 2(1), 2021.

[16] L. S. Bohannon, A. M. Herbert, J. B. Pelz, and E. M. Rantanen. Eye contact and video-mediated communication: A review. *Displays*, 34(2):177–185, 2013.

[17] P. Watzlawick, J. B. Bavelas, and D. D. Jackson. *Pragmatics of human communication: A study of interactional patterns, pathologies and paradoxes*. WW Norton & Company, 2011.

[18] S. Escalera et al. Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[19] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.

[20] L. Noy, E. Dekel, and U. Alon. The mirror game as a paradigm for studying the dynamics of two people improvising motion together. *Proceedings of the National Academy of Sciences*, 108:20947–20952, 2011.

[21] C. Leclère et al. Why synchrony matters during mother-child interactions: A systematic review. *PLoS ONE*, 9:e113571, 2014.

[22] E. Novotny and G. Bente. Identifying signatures of perceived interpersonal synchrony. *Journal of Nonverbal Behavior*, pages 1–33, 2022.

[23] F. T. Ramseyer. Motion energy analysis (mea): A primer on the assessment of motion from video. *Journal of Counseling Psychology*, 67:536–545, 2020.

[24] R. Feldman. Parent-infant synchrony: Biological foundations and developmental outcomes. *Current Directions in Psychological Science*, 16:340–345, 2007.

[25] R. Feldman. From biological rhythms to social rhythms: Physiological precursors of mother-infant synchrony. *Developmental Psychology*, 42:175, 2006.

[26] I. Gordon et al. Physiological and behavioral synchrony predict group cohesion and performance. *Scientific Reports*, 10:1–12, 2020.

[27] A. Murata, K. Nomura, J. Watanabe, and S. Kumano. Interpersonal physiological synchrony is associated with first person and third person subjective assessments of excitement during cooperative joint tasks. *Scientific Reports*, 11:1–11, 2021.

[28] J. A. Ward, D. Richardson, G. Orgs, K. Hunter, and A. Hamilton. Sensing interpersonal synchrony between actors and autistic children in theatre using wrist-worn accelerometers. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 148–155, 2018.

[29] R. C. Schmidt, S. Morr, P. Fitzpatrick, and M. J. Richardson. Measuring the dynamics of interactional synchrony. *Journal of Nonverbal Behavior*, 36:263–279, 2012.

[30] W. Pouw and J. A. Dixon. Gesture networks: Introducing dynamic time warping and network analysis for the kinematic study of gesture ensembles. *Discourse Processes*, 57:301–319, 2020.

[31] S. Keisari et al. Synchrony in old age: Playing the mirror game improves cognitive performance. *Clinical Gerontologist*, 45:312–326, 2022.

[32] S. M. Boker, J. L. Rotondo, M. Xu, and K. King. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods*, 7:338–355, 2002.

[33] I. Ravreby, Y. Shilat, and Y. Yeshurun. Liking as a balance between synchronization, complexity and novelty. *Scientific Reports*, 12:3181.

[34] M. Okano, M. Shinya, and K. Kudo. Paired synchronous rhythmic finger tapping without an external timing cue shows greater speed increases relative to those for solo tapping. *Scientific Reports*, 7:1–11, 2017.