

**RECOVERY OF DOMINANT SPATIOTEMPORAL
PATTERNS FROM MOVIES OF NEAR-SHORE TIDAL
WAVES**

Submitted by:

SAKSHAM KHETER PAL

Supervisor:

DR FINBARR O'SULLIVAN

Second Reader:

DR ERIC WOLSZTYNSKI



MSc Data Science and Analytics
School of Mathematical Sciences
University College, Cork

August 28, 2025

Abstract

There has been a massive increase in spatiotemporal datasets in today's world, Wave Datasets being one of them. Spatiotemporal analysis of wave video is essential for understanding ocean dynamics and pattern recognition. This study aims to apply Principal Component Analysis(PCA) to quantitatively analyze and reconstruct these wave video datasets. PCA was applied using a dominant color projection methodology to extract dominant spatial and temporal patterns effectively without a high computational load. PCA effectively extracted dominant patterns, but its linear nature limited reconstruction fidelity for highly complex videos. Overall, this study demonstrates a robust framework for quantitative analysis of spatiotemporal wave video datasets with potential applications in environmental monitoring and pattern recognition.

Declaration

I confirm that, except where indicated through the proper use of citations and references, this is my original work and that I have not submitted it for any other course or degree.

Signed: _____

Saksham Kheter Pal
August 28, 2025

Contents

Contents	iv
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Background and Related Work	2
2.1 Literature Review	2
2.2 Theoretical Background	5
2.2.1 Review of Spatiotemporal Data	5
2.2.2 Review of PCA and SVD	5
2.2.3 Mathematical Formulation of PCA	5
2.2.4 Mathematical Formulation of SVD	6
2.2.5 Intuition behind PCA and SVD	7
3 Methodology	8
3.1 Hardware and Software Specifications	8
3.1.1 Hardware Specifications	8
3.1.2 Development Environment	9
3.1.3 Software Libraries and Usability	9
3.2 Datasets and Data Collection	10
3.2.1 Small-Scale Dataset	10
3.2.2 Large-Scale Datasets	11
3.3 Preprocessing	12
3.3.1 Frame Extraction	12
3.3.2 Color Space Conversion	12
3.3.3 Normalization and Gamma Correction	12
3.3.4 Method-Specific Steps	12
3.4 Method 1: Full Principal Component Analysis on Video Data	13
3.5 Method 2: Dominant Color Projection	13
3.6 Summary	15

4 Experiments and Results on Wave Clip Datasets	16
4.1 Low Rank Decomposition of Small Scale Wave Clip	16
4.2 Summary of Key Observations for Small-Scale Wave clip	20
4.3 Large Scale Analysis on Wave Surface Data	20
4.4 Wave Clip 1	21
4.4.1 Low-Rank Decomposition on Wave Clip 1	21
4.5 Wave Clip 2	27
4.5.1 Low Rank Decomposition on Wave Clip 2	28
4.6 Summary of Key Observations for Large-Scale Wave Clips	33
5 Discussion	34
5.1 Overview of Quantitative Video Analysis	34
5.1.1 Comparison of Results between Small-Scale and Large-Scale Video Datasets	35
5.2 Effectiveness of PCA in spatiotemporal Video Analysis	36
5.3 Relation to Climatology and Environmental Monitoring	37
5.4 Applications and Broader Implications	38
5.5 Limitations of the Current Study	38
5.6 Future Work	39
6 Conclusion	40
A Key Code Blocks	41
A.1 PCA-Based RGB Channel Decomposition Code for Decomposition and Reconstruction for small wave clip	41
A.2 PCA Analysis on Projected Grayscale Video Derived from RGB Dominant Color Basis in Python	44
B Additional Resources	47
B.1 Code and Data Repository	47
B.2 Video Demonstrations	47
Bibliography	49

List of Tables

4.1	Cumulative variance explained at different PCA ranks.	16
4.2	Cumulative Variance Explained by Principal Components	21
4.3	Cumulative Variance Explained by Principal Components	28
5.1	Comparison of Key Dataset Characteristics	35
5.2	Comparison of Cumulative Variance Explained by Principal Components for Wave Clip 1 and Wave Clip 2	36

List of Figures

3.1 Separated Red, Green, and Blue channels (top row, left to right) from a single video frame, with the original reconstructed RGB frame shown in the bottom row.	10
4.1 Cumulative Variance explained by the top k -principal components	17
4.2 First three spatial modes extracted from Small-Scale Wave Clip. Each mode highlights areas of strong variance and recurrent structural features in the wave motion, revealing coherent spatial dynamics.	17
4.3 First 25 temporal modes extracted from the small-scale wave dataset. These components capture the primary patterns of temporal variation, providing insight into the most significant dynamic behaviors within the sequence.	18
4.4 Comparison of the original and PCA-reconstructed frames at time points $t = 5, 10, 20, 50$, and 100 for a reconstruction rank of $k = 5, 10, 20, 50$, and 100	19
4.5 Explained cumulative variance by number of components	22
4.6 First 16 spatial modes extracted from Wave Clip 1 using low-rank decomposition. These modes highlight areas of strong variance and recurrent structural features in the wave motion, revealing coherent spatial dynamics across the color spectrum. The patterns help identify how each mode contributes to the observed motion over time.	23
4.7 First three temporal modes from PCA of the wave video	24
4.8 Comparison of the original and reconstructed frames from Wave Clip 1 at different time points, using different numbers of singular value components ($k = 1, 10, 50$). The first column shows the original frames, while subsequent columns display reconstructions with increasing k values.	25
4.9 RGB intensity values of the center pixel over time in Wave Clip 1. The red, green, and blue channels exhibit fluctuating intensities that reflect local brightness changes due to wave motion. This signal offers insight into the temporal dynamics at a fixed spatial point and complements the global analysis captured by PCA.	26
4.10 Time series Plot of ten random pixels from Wave clip 1	27
4.11 Cumulative variance explained by the top k principal components for Wave Clip 2	28

4.12 First 16 Spatial PCA Modes	29
4.13 Temporal PCA modes for the first twenty-five spatial components of the projected wave video. Mode 1 remains dominant throughout most of the frames before tapering off.	30
4.14 Sample Frame Analysis: RGB, Projected, Mean, Std	31
4.15 Original vs Reconstructed Frames at Varying k Values	32
4.16 Center pixel RGB intensity over time for Wave Clip 2	32
4.17 Time series Plot of ten random pixels from Wave Clip 2.	33
5.1 Reconstruction error of the projected video as a function of PCA rank k . The mean squared error decreases with increasing rank, showing how the low-rank approximation captures the dominant spatial-temporal patterns. The right axis can optionally indicate the percentage error relative to the total variance.	37

Chapter 1

Introduction

In recent years, the widespread availability of high-resolution movie data has been on the rise, which has opened up new possibilities for analysis and studying complex physical systems through physical observations. This data serves us as a new rich source for quantitative information, being rich in both spatial and temporal information. From medical imaging to weather forecasting, the ability to process and analyze spatial-temporal data has become increasingly more valuable across scientific and engineering disciplines.

One such area of interest is the analysis of wave surface data, where the evolution of surface patterns over time can reveal important physical dynamics. For instance, understanding how water waves propagate, interact, and dissipate energy is essential in oceanography, coastal engineering, and climate modelling. The biggest challenge with such data, though, is the high dimensionality of it; each video captures hundreds of thousands of pixel values, and with hundreds or thousands of frames, the data quickly becomes difficult to analyze using traditional techniques. To address this, we employ multivariate statistical techniques, such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). These methods help us reduce dimensionality and extract underlying and meaningful patterns and dynamics hidden in the raw video data.

Chapter 2

Background and Related Work

2.1 Literature Review

Monitoring and understanding near-shore tidal wave dynamics is essential for coastal engineering, oceanography, sediment transport prediction, and hazard mitigation [Holman, Plant, and Holland 2013; Turner, Ranasinghe, and Short 2014]. These patterns can help us make observations about sediment budgets, predict ocean waves and currents, and morphological conditions. Such observations are also essential in other domains, including the military. [Holman, Plant, and Holland 2013]. As noted by Turner et al. [Turner, Ranasinghe, and Short 2014], these methods have been used in various locations to monitor and quantify the coastal response of the regional scale to sand nourishment and contribute to the construction of the first artificial reef on the Gold Coast. [Turner, Ranasinghe, and Short 2014]. Other useful applications of wave analysis include designing offshore wind turbines using the estimation of extreme loads, creating shore-break wave hazards forecast models to aid in public safety and lifeguard operations at beaches, and predicting erosion and sediment dynamics to aid shoreline management.[Dean, and Dalrymple 2004; Castelle et al. 2025; Pierella et al. 2020]

Several methods have previously been used to capture these patterns. Technologies such as UAVs, LiDAR, and satellite imagery have been utilized to understand coastal topography by examining coastal zones [Almeida, and R. Almar 2020; Rafael Almar et al. 2022; Viaña-Borja et al. 2024; Salameh et al. 2019]. These approaches provide high-resolution spatial and temporal data, enabling a detailed analysis of coastal processes. Several other foundational methods have previously been used to analyze wave data, including the Fast Fourier Transform (FFT) algorithm, which analyzes dominant spectrum features using the Fourier transform, and wavelet analysis, which uses the wavelet transformation to analyze ocean surface wave records. [J. Chen 2018; Massel 2000]. Other hardware-based methods that have been researched include using drones and photogrammetry for analysing sand beach topography[Casella et al. 2020]. Although these methods have proven to be useful, they do have some limitations. While remote sensing methods provide valuable data, LiDAR can be cost-prohibitive, limiting its

application. Another limitation of these methods is not being able to capture fine-scale patterns.

While these classical methods do provide reliable measurements, modern spatiotemporal datasets require advanced analytical techniques. Consequently, several new mathematical methodologies and techniques have been explored to analyze modern spatio-temporal data. The Video Covariance Matrix Logarithm (VCML) proposed by Bilinski and Bremond [Bilinski, and Bremond 2015] introduces a covariance matrix-based descriptor for human action recognition in videos, capturing local spatiotemporal features. Another approach by Wang et al. [Wang et al. 2014] employs low-rank matrix estimation for spatio-temporal image reconstruction in dynamic photoacoustic computed tomography, leveraging temporal correlations to enhance image quality. There exist several studies that use tensor decomposition techniques for the analysis of spatiotemporal data decomposition [X. Chen et al. 2020; Gong, Huang, and Yang 2023; Yan et al. 2021]. Another method that has been constantly used and associated with spatio-temporal structures is Dynamic Mode Decomposition, which extracts patterns from time series data. Building upon this, TDMD(Tensor Dynamic Mode Decomposition), which is an extension of Dynamic Mode Decomposition, handles multidimensional tensor data, preserving both spatial and temporal structures without flattening and compromising on the quality [He et al. 2025]. Other methods include STVNN, which refers to spatiotemporal covariance neural networks, models spatiotemporal interactions in multivariate time series[Cavallo, Sabbaqi, and Isufi 2024]. Building on these approaches, Principal Component Analysis(PCA) and Singular Value Decomposition(SVD) provide a strong framework for extracting dominant patterns from spatio-temporal data.

Principal Component Analysis (PCA) and Singular Value Decomposition are foundational multivariate statistical methods that have been extensively researched and applied in multivariate statistics, proven to be the most cost-effective and efficient method for extracting spatiotemporal patterns from video datasets. PCA was first introduced by Karl Pearson in 1901 and later formalized in multivariate statistics by Hotelling. PCA and SVD work on the methodology of dimensionality reduction of high-dimensional data to make it feasible for analysis. PCA identifies orthogonal directions of maximum variance, while SVD decomposes datasets into orthonormal spatial and temporal modes - often used for compression and noise filtering. SVD is often employed as the primary means for computing PCA [Gyimadu, and Bell 2025].

PCA has been used excessively on multiple kinds of spatiotemporal data. In the field of neuroscience, PCA has been utilized to analyze brain activity data and identify latent structures and the underlying dynamics of cognitive processes [Brown et al. 2022]. Other implementations in neuroscience include dPCA (Demixed Principal Component Analysis), which is a form of PCA that decomposes neural population activity into components representing different experimental factors, facilitating the interpretation of complex neural data[Kobak et al. 2016], and Bayesian functional PCA, which is a PCA model that accounts for the complex spatiotemporal nature of group-level brain activity data[Margaritella, Inacio, and King 2023].

Beyond Neuroscience, PCA has been used in Traffic and Urban Studies, to predict

traffic flow, incorporating PCA in models for dimensionality reduction [Ding et al. 2025; Caffo et al. 2010]. In climate and environmental science, for the assessment of climate change and other sectors of climatology [Tadic, Bonacci, and Brleković 2019].

While PCA and SVD have been extensively used in a lot of high-resolution spatiotemporal datasets, they come with certain limitations. PCA often includes linear relationships between variables, so it may miss nonlinear patterns. It may also become computationally expensive for very large matrices and difficult to interpret in terms of the original features.

Several methodologies have been developed to work on more complex datasets using non-linear PCA. One such example is Rock-PCA, which is a kernel-based nonlinear dimensionality technique used to analyze spatio-temporal earth system data [Bueso, Piles, and Camps-Valls 2020]. STPCA is a two-stage SVD method to extract patterns from fMRI data, and Zoom SVD. [Jang et al. 2018; Caffo et al. 2010; Krzyżko et al. 2023]. Several more Nonlinear methods of PCA have been used for a multitude of datasets, including the analysis of global soil moisture and implementation on the Lorenz System [Bueso, Piles, and Camps-Valls 2020; Monahan 2000]. While Non-Linear PCA is usually implemented to get over the limitation of linearity in regular PCA, Research has shown that Non-Linear PCA has often led to spurious bimodality in the field of atmospheric sciences and is therefore error-prone, which raises the question of its reliability and highlights the need for careful implementation and validation [Christiansen 2005].

Principal Component Analysis has been successfully applied in wave and coastal areas. For instance, Mori et al successfully used PCA for wave climate projection, which is important for climate impact assessment[Kishimoto et al. 2017]. Similarly, Coastal Wave modelling efforts have integrated PCA to reduce dimensionality [Camus et al. 2013]. Several studies have adopted a hybrid methodology, combining PCA-based dimensionality reduction with data-driven techniques to analyze coastal regions, wave patterns, storm-surge responses, and surf-zone dynamics [Camus et al. 2013; Ricondo, MÁC̄ndez, et al. 2023; Naeini, and Snaiki 2024]. FPCA(Functional PCA) has been used for sensitivity analysis of high-dimensional coastal flood simulators, dealing with issues such as non-stationary spatial outputs and high computational time[Perrin et al. 2020]. However, while these studies demonstrate the utility of PCA for dimensionality reduction and hybrid modelling in wave datasets, they primarily emphasize large-scale projections and model performance.

While PCA and SVD have been used extensively with spatiotemporal datasets, their applications to high-resolution spatiotemporal wave data still remain largely unexplored. Existing studies focus primarily on large-scale climate projections or dimensionality reduction for wave modelling. However, these applications generally treat PCA as a pre-processing tool rather than a primary means of extracting interpretable spatiotemporal patterns. As a result, little relevance has been given to the direct quantitative analysis of coastal wave video data, particularly in terms of data such as variance explained or temporal patterns. This gap motivates the present study, which applies PCA and SVD directly to high-resolution wave video datasets in order to identify underlying dynamics and conduct a thorough quantitative analysis.

2.2 Theoretical Background

This chapter presents the theoretical foundation necessary to understand the techniques used in this study. It begins by defining spatiotemporal data, a key concept used in video data and analysis. We then review Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), both essential techniques in multivariate analysis of spatiotemporal data.

2.2.1 Review of Spatiotemporal Data

Spatiotemporal data literally refers to data that varies over both space and time. In the context of the movie data we are working with, each frame captures the spatial configuration of a system, while the sequence of frames shows how the configuration changes over time. Spatiotemporal data comprises the following components:

- **Spatial component:** arrangement of pixel intensities in each frame.
- **Temporal component:** evolution of these spatial patterns over time.

This makes wave videos a natural example of spatiotemporal data. Waves are inherently time-varying phenomena â their shape and intensity evolve continuously, influenced by factors such as wind, current, and boundary conditions.

As highlighted by Qin Y [2020] in their study of large-scale ocean forecasting data, the complexity and volume of such datasets necessitate the use of sophisticated tools capable of distilling essential information. Techniques like PCA and SVD are particularly effective here, offering a way to decompose spatiotemporal data into dominant modes that represent recurring patterns or movements.

2.2.2 Review of PCA and SVD

To understand and move forward with the analysis conducted in this project, we first review the mathematical foundations and key concepts of Principal Component Analysis and Singular Value Decomposition. PCA is one of the oldest and most widely used techniques for dimensionality reduction. Its primary goal is to reduce the number of variables in a dataset while preserving as much of the original variance as possible. This is achieved by finding new variables called principal components, which are linear combinations of the original variables that maximize variance and are mutually uncorrelated (orthogonal).

2.2.3 Mathematical Formulation of PCA

Given a random vector $\mathbf{Y} \in \mathbb{R}^P$ with mean vector $\boldsymbol{\mu} = \mathbb{E}[\mathbf{Y}]$ and covariance matrix $\Sigma = \text{Var}(\mathbf{Y}) = \Gamma \Lambda \Gamma^\top$, where Γ is the matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues sorted in descending order, PCA performs an orthogonal linear transformation as follows:

$$\mathbf{X} = \Gamma^\top (\mathbf{Y} - \boldsymbol{\mu})$$

The original vector \mathbf{Y} can be approximated using the first K principal components as:

$$\mathbf{Y} \approx \boldsymbol{\mu} + \sum_{j=1}^K \boldsymbol{\gamma}_j X_j$$

where $\boldsymbol{\gamma}_j$ are the loading vectors (columns of Γ) and X_j are the corresponding principal component scores.

2.2.4 Mathematical Formulation of SVD

Given a real matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the Singular Value Decomposition (SVD) of \mathbf{X} factorizes it into three matrices:

$$\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^\top$$

where:

- $\mathbf{U} \in \mathbb{R}^{m \times n}$ contains the *left singular vectors* of \mathbf{X} ,
- $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a diagonal matrix of singular values,
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ contains the *right singular vectors* of \mathbf{X} ,
- and \mathbf{V}^\top is the transpose of \mathbf{V} .

The columns of \mathbf{U} , denoted by $\boldsymbol{\mu}_i$, form an orthogonal basis for the column space of \mathbf{X} such that:

$$\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

The decomposition can also be expressed as:

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$$

where \mathbf{D} is a diagonal matrix with singular values $d_1 \geq d_2 \geq \dots \geq d_p \geq 0$, sorted in descending order. This enables us to write \mathbf{X} as a sum of rank-one matrices:

$$\mathbf{X} = d_1 \mathbf{u}_1 \mathbf{v}_1^\top + d_2 \mathbf{u}_2 \mathbf{v}_2^\top + \dots + d_p \mathbf{u}_p \mathbf{v}_p^\top$$

Each term $d_k \mathbf{u}_k \mathbf{v}_k^\top$ captures the contribution of the k -th mode to the overall structure of the data matrix \mathbf{X} . This formulation is particularly useful in applications such as dimensionality reduction, noise filtering, and image or video reconstruction, as it allows approximation of \mathbf{X} by retaining only the first k dominant terms.

2.2.5 Intuition behind PCA and SVD

From the above mathematical formulation of PCA and SVD, we can infer that the derivation of PCA leads to a sequential algorithm that applies eigendecomposition to a covariance matrix derived from the data. However, a challenge arises when the data dimensionality is high, requiring the application of SVD to the data matrix. This is typically done after subtracting the mean location from the data, a step known as *mean centering* [Park, and Bae 2023].

To summarize, PCA helps us find new orthogonal axes (principal components) ordered by the amount of variance they explain. Geometrically, it rotates the data to align with directions of highest variability, which are identified via the eigenvectors of the covariance matrix. SVD, on the other hand, is a more general decomposition technique â in fact, PCA is essentially SVD applied to a mean-centered data matrix.

In the context of our spatiotemporal wave clips, SVD allows us to decompose the data into spatial and temporal-spatial patterns. By reconstructing our dataset using only the top k components, we can effectively denoise or compress the signal. The first mode typically captures the dominant movement or oscillation, while subsequent modes describe finer structure or noise.

Chapter 3

Methodology

This chapter outlines the methodological frameworks used for analyzing the spatiotemporal wave-clip datasets. This chapter discusses the preprocessing pipeline and the two different methodologies implemented in this study for different types of datasets. The two complementary approaches were

1. A baseline full-frame Principal Component analysis method applied directly to the video
2. A single-color projection pipeline, which reduces dimensionality at the preprocessing stage by projecting the video onto its dominant color component before spatial PCA.

Both methods share common preprocessing steps, data representation, and evaluation metrics, but differ in how they capture spatial-temporal variability.

3.1 Hardware and Software Specifications

This section showcases the hardware specifications the analysis was performed, and the programming languages and libraries that were subsequently used on it.

3.1.1 Hardware Specifications

The experiments were conducted on a personal workstation with the following specifications:

- CPU: Intel i5-12450HX, 12 cores, 2.4 GHz
- RAM: 16GB DDR4
- GPU: NVIDIA RTX3060 Laptop GPU(6GB VRAM)
- Operating System: Windows 11 64-bit
- Storage: 512GB SSD

3.1.2 Development Environment

The analysis was conducted using RStudio (Version 2024.04.2) as the primary IDE for R scripting, debugging, and plotting. For Python scripts, VS Code (version: 1.103.1) was used, providing flexible code editing, debugging, and visualization capabilities. Both environments facilitated reproducible workflows and streamlined pipeline execution.

3.1.3 Software Libraries and Usability

The analysis pipeline for this analysis makes use of a combination of both Python and R libraries to handle preprocessing tasks, visualization, Image manipulation, and dimensionality reduction.

In R, the primary libraries used are `av`, `magick`, and `png`, while Python usually relies on libraries such as `OpenCV`, `Numpy`, and `Matplotlib`.

The `av` package in R is used for the extraction of individual frames from videos(.mp4). This allows each frame to be extracted individually for analysis. The `magick` library facilitates image manipulation, including resizing, rotation, channel separation, and conversion of images to numerical matrices suitable for mathematical operations. The `png` library is used to read and write PNG images when precise control over pixel intensities is required, enabling accurate reconstruction and visualization of frames after processing. Together, these R libraries enable a complete workflow from frame extraction, channel separation, PCA-based dimensionality reduction, and frame reconstruction back into images and video clips.

Figure 3.1 showcases the ability of the `av` and `magick` libraries in R to read RGB matrices from a single .jpeg image and reconstruct it back into the original image, and subsequently using the array of images obtained to convert it back into video format(.mp4). The `magick` package is used to isolate the three isolate color channels from any given .jpeg image and convert it into a numeric array representing pixel intensities in each channel. To reassemble these back the `magick` package encodes these reconstructed arrays back into .jpeg format. This workflow demonstrates how individual color channel data can be manipulated and visualized, providing a foundation for subsequent dimensionality reduction (e.g., PCA) and pattern extraction analyses.

The reassembled .jpeg frames can finally be converted back into a video clip (.mp4 file). This method of frame extraction, channel separation, reconstruction, and video reassembly illustrates the capabilities of R's multimedia libraries in supporting advanced spatiotemporal data analysis for environmental video datasets.

In Python, on the other hand, `OpenCV` is used for fast frame extraction and color separation, providing optimized routines for reading, resizing, and color separation. `Numpy` is primarily used for numerical operations such as flattening of frames or computing PCA. `Matplotlib` provides visualization capabilities for temporal and spatial patterns, variance explained plots, and frame comparisons. These Python libraries offer efficient memory management, vectorized computations, and optimized routines for high-resolution or long-duration video processing, enabling faster frame extraction, dimensionality reduction, and reconstruction without running into performance

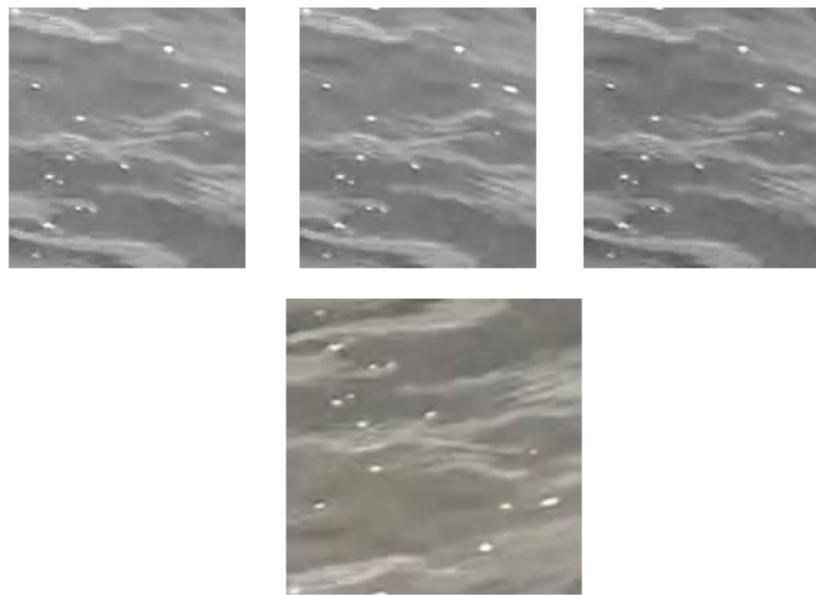


Figure 3.1: Separated Red, Green, and Blue channels (top row, left to right) from a single video frame, with the original reconstructed RGB frame shown in the bottom row.

bottlenecks, making these libraries perfect for higher-dimensional dataset analysis

Overall, these libraries were selected for their usability, performance, and interoperability. By combining R and Python tools for different datasets, the workflow is both robust and adaptable to different video formats, resolutions, and analytical requirements.

3.2 Datasets and Data Collection

The primary datasets for this study consist of clips of waves from different bodies of water captured at different locations, each having distinct characteristics and complexity, to obtain a diverse set of results for our analysis.

3.2.1 Small-Scale Dataset

The small-scale dataset consists of a short video clip with reduced spatial and temporal resolution. For this part of the project, we are using a wave clip of dimensionality 140 x 154. This video runs at 30 frames per second with a total number of 1790 frames. The video file is in .mp4 format, which is widely used for efficient storage and playback of high-quality video. This is a spatiotemporal wave clip due to its evolving nature over time, i.e., the movement of the wave over time, which is what makes this video well-

suit for dimensionality reduction techniques, as we can observe how dominant spatial features change across frames. The clip doesn't have much complexity and follows a regular oscillatory motion with little to no sunlight or uncertain wave conditions. This clip serves as a simple, small-scale dataset to apply a full-scale PCA methodology to. The original video can be accessed by clicking the following link:

[Click here to watch the original small-scale video clip](#)

3.2.2 Large-Scale Datasets

The large-scale dataset consists of high-definition (1920×1080 pixels) video clips of ocean surface waves recorded at 30 frames per second, with each clip lasting approximately two minutes. These datasets were used for the main analysis due to their ability to capture realistic wave dynamics at a higher level of detail. Video stabilization and resolution enhancement were applied using post-processing software to improve the effectiveness of PCA analysis. Several recent studies support the feasibility of using consumer-grade video systems for robust wave analysis. For instance, Vivera et al. (2020) employed synchronized smartphone video to estimate directional wind wave spectra with accuracy comparable to traditional instruments. Similarly, Ondoa et al. (2019) demonstrated the effectiveness of video-sensed wave characteristics for shoreline water level estimation. These studies support the use of high-definition consumer footage as a practical and insightful real-world dataset for low-rank decomposition of wave dynamics.

Description of Large Scale Wave Clip 1

For our first example, we captured a video using an iPhone 16 at 30 frames per second and a resolution of 1920×1080 (Full HD). The recording was done around the evening at a lakeside location in New York, where I personally set up a stable tripod to maintain a fixed viewpoint. The video was taken under low natural lighting conditions with minimal direct sunlight and low wind. This setting provides a relatively calm wave pattern, making it ideal for analyzing spatiotemporal motion. The original Wave Clip 1 can be accessed by clicking the following link:

[Click here to watch the original large-scale video clip 1](#)

Description of Large Scale Wave Clip 2

For our second example, we captured a video using an iPhone 16 at 30 frames per second and a resolution of 1920×1080 (Full HD). The recording was done in the afternoon at a lakeside location in Cork, where I personally set up the camera on stable tripod to ensure a fixed viewpoint. The video was taken under natural lighting with sunlight in some frames and low wind conditions. The wave pattern is constant with some small deviations, which makes it perfect to extract spatiotemporal patterns. The original Wave Clip 2 can be accessed by clicking the following link:

[Click here to watch the original large-scale video clip 2](#)

3.3 Preprocessing

This chapter aims to explain the steps used for preprocessing of the datasets before applying dimensionality reduction and pattern extraction. The raw video data underwent numerous steps to reduce noise, ensure consistency, and make the data suitable for PCA.

3.3.1 Frame Extraction

Each input video is decomposed into individual frames using standard video processing libraries in Python and R. For high-resolution datasets (1920x1080), the frames were downsampled to 256x144 to reduce computational load. This ensures the analysis can be performed without running into any memory issues.

3.3.2 Color Space Conversion

All frames are converted to the RGB color space to allow channel-wise analysis. Each frame is represented as a three-dimensional array, with layers corresponding to the three colors Red, Green, and blue channels. This representation is valid for both approaches.

3.3.3 Normalization and Gamma Correction

Pixel intensities are normalized to the range [0, 1] and optionally subjected to mild gamma correction ($\gamma = 1.8$) to improve clarity. This step helps reduce color distortion and ensures PCA works on comparable scales, and makes sure the output frames are visually interpretable.

3.3.4 Method-Specific Steps

Full-Frame PCA Pipeline : For the full-frame PCA method, each frame is flattened into a single vector by converting the RGB channels into a single array by concatenating all channels sequentially. This produces a two-dimensional matrix with dimensions $N_f \times (H \cdot W \cdot 3)$, where N_f is the number of frames and H and W are the frame height and width, respectively. This matrix serves as the input for PCA decomposition.

Dominant Color Projection Pipeline: In the dominant color projection pipeline, each frame is projected onto its dominant color component before flattening. Specifically, the primary color axis is extracted using a preliminary PCA on the RGB channels, and each frame is projected along this axis to generate a single-channel representation. These projected frames are then treated as grayscale images and concatenated across frames to form the input matrix for subsequent spatial-temporal PCA. This approach emphasizes the principal color variation across frames while reducing data dimensionality, allowing efficient extraction of temporal and spatial patterns.

3.4 METHOD 1: FULL PRINCIPAL COMPONENT ANALYSIS ON VIDEO DATA

3.4 Method 1: Full Principal Component Analysis on Video Data

Full Principal Component Analysis(PCA) was applied to the small video dataset to reduce dimensionality while preserving its dominant spatio-temporal structures.

After the preprocessing steps, as described in section 4.2, each video frame was flattened into a single vector by concatenating the RGB channels, producing a data matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_{N_f}^\top \end{bmatrix} \in \mathbb{R}^{N_f \times (H \cdot W \cdot 3)}$$

where N_f is the number of frames, and H and W are the frame height and width, respectively.

Full PCA was then performed on the centered data matrix to obtain the Principal Components:

$$\mathbf{X}_c = \mathbf{T}\mathbf{P}^\top,$$

Where $\mathbf{T} \in \mathbb{R}^{N_f \times k}$ represent the scores of the frames in the reduced k -dimensional space, and $\mathbf{P} \in \mathbb{R}^{(H \cdot W \cdot 3) \times k}$ are the loading vectors.

The cumulative variance explained by the first k components was calculated as:

$$\text{Variance Explained (Cumulative)} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^{H \cdot W \cdot 3} \lambda_i} \times 100\%,$$

For reconstruction, each frame was approximated using the top components k :

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + \sum_{j=1}^k t_{ij} \mathbf{p}_j,$$

where t_{ij} is the score of frame i on component j , and \mathbf{p}_j is the corresponding loading vector. Reconstructed frames were then reshaped back into the original spatial dimensions with RGB channels and saved for visual inspection and variance analysis.

All calculations were performed in R using the `av` and `magick` libraries and the `prcomp` function, with an optional gamma correction applied to improve color fidelity and ensure comparability across frames.

3.5 Method 2: Dominant Color Projection

The dominant color projection method is designed to handle high-dimensional and more complex video datasets efficiently, and reduce computational time and load

while also preserving spatial and temporal patterns. Each frame is projected along its dominant color axis, producing a single-channel representation for spatial-temporal PCA. This reduces the data size from $H \cdot W \cdot 3$ to $H \cdot W$ per frame, significantly reducing the computational load.

All the RGB pixel values across the video are first centered and normalized using the equation:

$$\mathbf{c}_{i,j,t}^{\text{norm}} = \frac{\mathbf{c}_{i,j,t} - \bar{\mathbf{c}}}{\sigma_{\mathbf{c}}},$$

After which, we apply PCA on the resulting color correlation matrix to obtain the dominant color direction. Each pixel is projected along this axis:

$$f_{i,j,t} = (\mathbf{c}_{i,j,t} - \bar{\mathbf{c}})^{\top} \mathbf{p}_{\text{dom}},$$

producing a grayscale video.

The projected frames are standardized across time to remove pixel-wise biases using:

$$f'_{i,j,t} = \frac{f_{i,j,t} - \mu_{i,j}}{\sigma_{i,j}},$$

and flattened spatially to form the matrix

$$\mathbf{Y} \in \mathbb{R}^{(H \cdot W) \times T},$$

PCA is then applied to \mathbf{Y} using:

$$\mathbf{Y}_c = \mathbf{Y} - \bar{\mathbf{Y}} = \mathbf{T} \mathbf{P}^{\top},$$

where \mathbf{T} represents temporal coefficients and \mathbf{P} the spatial modes

Finally, reconstruction of the top k components is applied as:

$$\hat{\mathbf{f}}_t = \bar{\mathbf{f}} + \sum_{j=1}^k t_{tj} \mathbf{p}_j,$$

which is then back-projected to RGB via the dominant color axis:

$$\hat{\mathbf{c}}_{i,j,t} = \hat{f}_{i,j,t} \mathbf{p}_{\text{dom}} + \bar{\mathbf{c}}.$$

This approach is applied primarily to larger, more complex, or high-resolution video data using Python. This method prevents the system from running into any memory usage issues and makes the analysis actually feasible. Python is used due to its efficiency in handling high-dimensional video data using libraries such as numpy and OpenCV, allowing seamless integration for preprocessing and PCA computations.

3.6 Summary

Two PCA-based approaches were explored in this section, both designed to extract spatial and temporal patterns from different types of wave clips. The first, full-frame PCA, flattens each RGB frame and then applies PCA directly to capture dominant modes. The second approach projects frames onto a dominant color axis before applying PCA, creating a single channel, reducing the computational load, and making it suitable for analysis of larger datasets. Both methods incorporate reconstruction, quantitative analysis, and visualizations to assess variance, fidelity, and spatiotemporal dynamics, providing an efficient and interpretable pipeline. The following chapter puts these methods to use and interprets the results and visualizations obtained from them to obtain meaningful insights.

Chapter 4

Experiments and Results on Wave Clip Datasets

4.1 Low Rank Decomposition of Small Scale Wave Clip

To provide an intuitive understanding of the PCA and SVD frameworks, we apply these methods to a short video of near-shore tidal wave motion. The dataset consists of a video sequence capturing near-shore tidal waves, recorded at 30 frames per second. Each frame, initially of shape (height x width), is flattened into a vector of pixel intensities. These vectors are then arranged column-wise into a data matrix $\mathbf{X} \in \mathbb{R}^{p \times t}$, where p is the number of pixels per frame and t is the number of frames.

PCA is performed on the mean-centered matrix \mathbf{X} to extract dominant spatiotemporal patterns. Reconstructions at varying PCA ranks k were generated, and other analyses were done to extract spatial and temporal patterns from the clip.

Rank (Number of PCs)	Cumulative Variance Explained (%)
1	10.54
5	29.32
10	39.16
20	51.04
50	66.09
100	76.27

Table 4.1: Cumulative variance explained at different PCA ranks.

Figure 4.1 and Table 4.1 display the cumulative variance explained at different values of k (Number of Components). We can interpret from the plot that as the number of components increases, so does the variance explained. There is a significant jump in the variance explained from small to large k (up to 200). A high value, such as 50% at

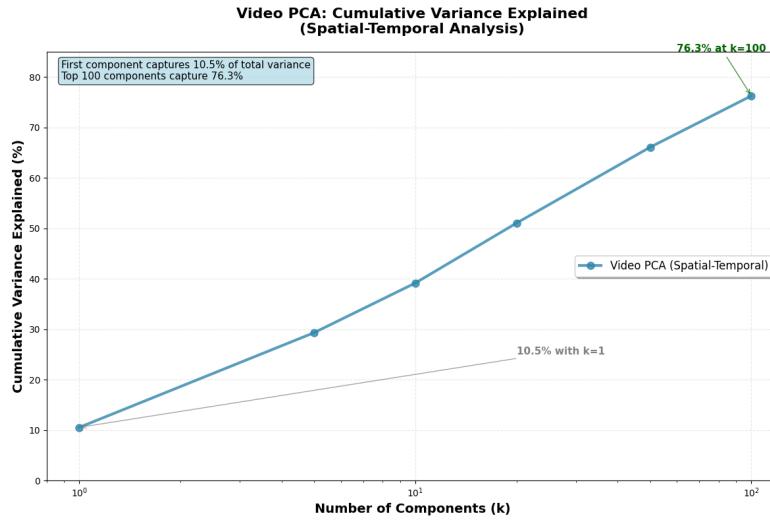


Figure 4.1: Cumulative Variance explained by the top k-principal components

just 25 components, indicates that most of the dominant spatial and temporal patterns could easily be captured at a small number of modes. To further examine this, we now analyse the leading spatial modes that contribute strongly to this explained variance.



Figure 4.2: First three spatial modes extracted from Small-Scale Wave Clip. Each mode highlights areas of strong variance and recurrent structural features in the wave motion, revealing coherent spatial dynamics.

The spatial modes in figure 4.2 highlight coherent wave structures across the field. Looking at the figure, we can interpret important patterns. The first mode is the dominant spatial gradient accounting for at least 10.5 % of the variability. The diagonal brightness contrast might suggest a repeating structure in the wave field

The second spatial mode is responsible for 6.6% of the variance, indicating a rapid spatial intensity structure. This mode likely suggests wave activity mostly in the middle with the calmer edges.

The third mode captures 4.1% of variance, capturing mostly finer-grained variations.

The figure has small, darker regions with brighter sides, not uniform but localized, maybe suggesting wave crests forming in the middle region.

Together, these three modes explain about 21.3% of the spatial variance, implying there's no single mode that dominates strongly, suggesting the variability is spread across multiple spatial modes and the wave field, even after having low dimensionality(140x154), is still rather complex. To get a more complete understanding, we now move towards analysing the temporal modes of the clip next.

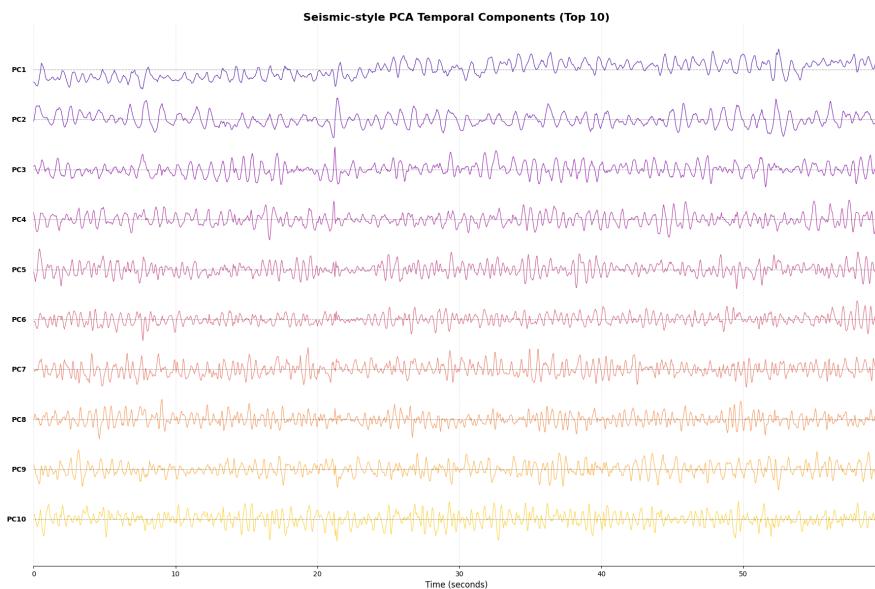


Figure 4.3: First 25 temporal modes extracted from the small-scale wave dataset. These components capture the primary patterns of temporal variation, providing insight into the most significant dynamic behaviors within the sequence.

Looking at figure 4.3, we can examine the temporal dynamics of the wave clip. PC1 seems to be the most dominant mode, capturing most of the temporal variation in the clip. The next 2 modes, PC2 and PC3, display medium frequency variations with some periodic structure. PC2 shows more regular behavior than PC1. These modes capture secondary wave motions.

As the number of modes keeps increasing, we start seeing increasingly more complex patterns. More irregular oscillations can be seen through PC4 to PC6, and noise-like patterns are observed from PC7 to PC10. These modes suggest more irregular motions. This clear progression from smooth to complex fine-scale dynamics suggests a hierarchical wave structure. This also suggests the wave field has coherent dynamics across multiple scales.

To see how these temporal and spatial patterns appear in the wave field, we now compare the original frames with their PCA reconstructions at selected time frames. The original video plotted against reconstructed videos at different values of k can be accessed by clicking the link below:

[Click Here to watch the Original Clip Plotted against Reconstructed Clips at different values of \$K\$](#)

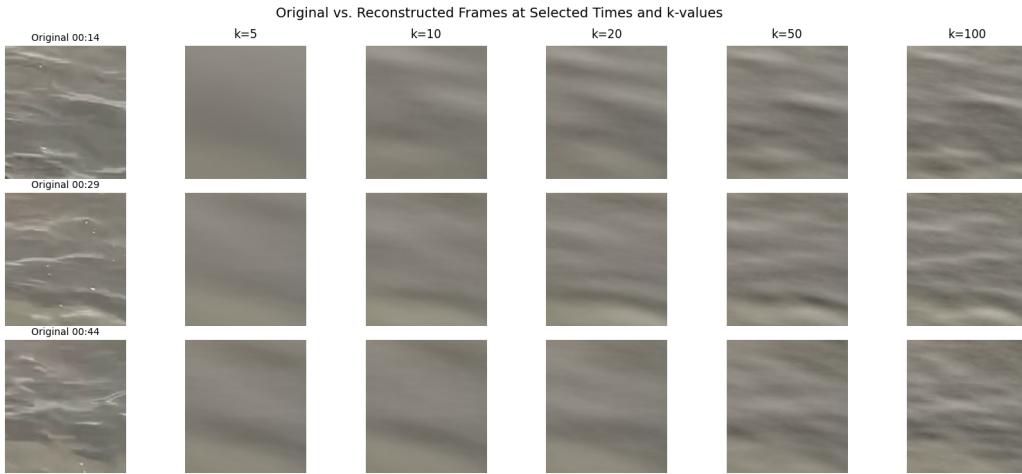


Figure 4.4: Comparison of the original and PCA-reconstructed frames at time points $t = 5, 10, 20, 50$, and 100 for a reconstruction rank of $k = 5, 10, 20, 50$, and 100 .

We created multiple reconstructed video sequences using different numbers of singular value components (denoted as k), ranging from 1 to 100. These reconstructions offer a visual understanding of how much of the original dynamics can be retained using only a few dominant spatial and temporal modes. Similar strategies have proven successful in related spatiotemporal domains. For instance, J. Zhang, Laufer, and Beard [2014] demonstrated how low-rank matrix estimation enables efficient reconstruction of dynamic biomedical images, underscoring the broader applicability of such models in preserving essential dynamics across disciplines. This experiment illustrates the core idea behind dimensionality reduction and spatiotemporal decomposition. By applying PCA to complex natural video data, we can separate coherent wave structures from noise and extract meaningful patterns.

By reconstructing videos using a limited number of singular value components, we achieve a balance between data compression and the preservation of key motion features – a crucial benefit when dealing with large datasets where computational efficiency is important.

The video reconstructed with $k = 1$ is nearly static, lacking visible motion or detailed structure, as it only captures the most dominant spatial pattern. This demonstrates that a single low-rank mode is insufficient to capture the complexity of a spatiotemporal wave clip.

At $k = 5$, more wave motion and minute spatial features begin to show, though the video remains vague to draw meaningful conclusions. At $k = 10$, the reconstruction starts becoming closer to the original. Dominant structures, such as wave lines and patterns. Similar Visual patterns are observed at $k = 20$. At $k = 50$, the reconstruction

successfully captures subtle textures and spatial changes, and by $k = 100$ it starts becoming visually closer to the original. Although the reconstructed video does look visually similar to the original at this point, to capture finer details and conduct more nuanced analysis, even higher modes are required.

The original raw video and reconstructed versions at different k values can be accessed through the link in the Appendix.

4.2 Summary of Key Observations for Small-Scale Wave clip

This section of the study further reinforces the utility of low-rank reconstruction for spatiotemporal data using different values of k , the number of retained singular value components, to capture dominant spatial patterns. By systematically varying k , we were able to analyze the trade-off between compression and reconstruction fidelity.

At the lowest rank ($k = 1$), the reconstruction captures only the most dominant static spatial structure, resulting in a near-static approximation of the video. As k increases to intermediate values such as $k = 5$ or $k = 10$, the reconstructed sequences begin to reveal wave motion and more nuanced spatial patterns. At higher ranks, reconstructions achieve significantly better fidelity and become visually similar to the original video. This validates the effectiveness of using additional modes to preserve critical dynamic and structural information.

However, reconstructions at higher k values also exhibit minor limitations, such as difficulty capturing abrupt intensity spikes and occasional overestimation of pixel values in localized regions. Despite these challenges, the overall experiment strongly supports the effectiveness of PCA and SVD for analyzing high-dimensional spatiotemporal datasets.

4.3 Large Scale Analysis on Wave Surface Data

In this section of the study, we dive deeper into larger-scale wave surface datasets to retrieve dominant spatiotemporal patterns. Specifically, we focus on two high-resolution videos captured in real-time during the course of this research. As the dataset size increases, so does the number of frames, resulting in a corresponding increase in dimensionality and computational demands. While our previous experiments successfully demonstrated insights such as optimal values of k for reconstruction and the temporal independence of modes, these techniques become less scalable and efficient when applied to longer, high-resolution clips.

Nevertheless, larger wave surface datasets offer a valuable opportunity for richer spatial and temporal analysis. For instance, Duan, Li, and L. Zhang [2024] emphasizes that analyzing extensive wave data, especially involving variations in wave height and surface motion, can support applications ranging from marine ecological assessments

to ship navigation safety and meteorological forecasting. Given that over 70% of Earth's surface is covered by water, understanding these patterns is also critical for modeling air-sea interactions and developing high-resolution climate models [Liang, Xu, and R. Zhang 2022].

Consequently, we propose an adapted approach that maintains the core principles of dimensionality reduction while introducing optimizations for scalability and performance. This revised method uses dominant color projection PCA, outlined previously in the methodology section of the study, and analyzes how this new methodology scales with larger and more complex datasets. The following sections of the study analyze these larger datasets and make meaningful interpretations about their spatial and temporal patterns observed.

4.4 Wave Clip 1

This Section explores the preprocessing and analysis of the first wave clip captured in real time during this study. The aim is to extract and analyze dominant spatiotemporal patterns from this video using the methodology defined in the previous section and later compare its results with other video clips in the study.

4.4.1 Low-Rank Decomposition on Wave Clip 1

We apply the previously discussed PCA formulation to extract dominant spatiotemporal patterns from the RGB channels. Reconstruction of the video was performed for $k = 1, 10, 20, 50, 100, 200$. Due to computational limitations, we restricted our analysis to $k \leq 200$.

Table 4.2: Cumulative Variance Explained by Principal Components

Components (k)	Temporal Correlation PCA (%)	V-Domain PCA (%)
1	10.30	10.16
5	35.36	36.37
10	47.56	49.28
20	57.94	59.44
50	73.00	73.88
100	84.02	84.49
200	92.30	92.52
500	97.26	97.36
1000	98.68	98.73

The explained variance plot (figure 4.5 and table 4.2) reveals important patterns about the spatial and temporal structure of the wave clip.

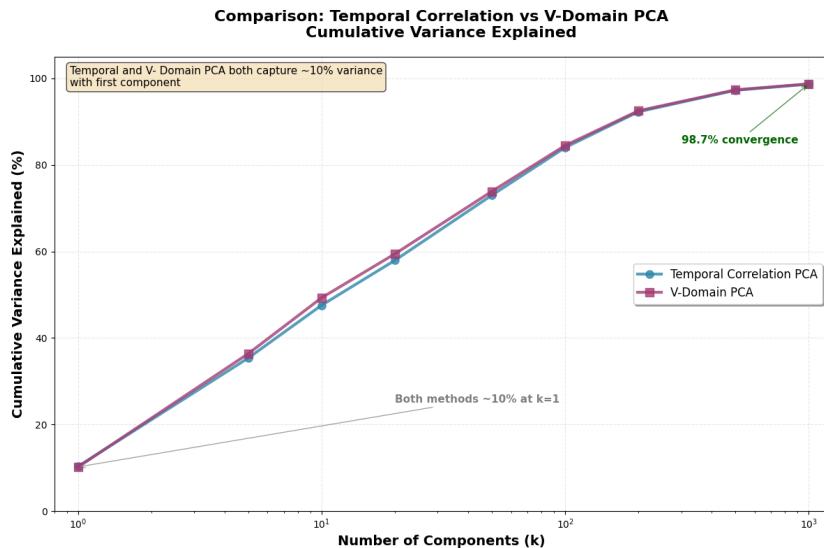


Figure 4.5: Cumulative variance explained as a function of the number of principal components retained.

We obtain almost the same variance for both the correlation matrix of the temporal data and the projected data, implying that there are no dominant temporal patterns to be obtained in this clip, and the spatial content is relatively homogeneous.

Observing the values more clearly, by $k=100$, almost 84% of the variance has been explained both temporally and spatially, implying that the differences are almost negligible at most k -values. Although the variance explained does keep increasing after this point, the jump isn't substantial. Most variance captured after this point is finer spatial details. While $k=1000$ achieves near-perfect variance explained, lower rank approximations already capture the majority of the meaningful motion, which may be preferable in contexts requiring compression.

These results suggest that this video is spatially and temporally uniform because there's not much dominant structure in either dimension. The video has balanced, uniform characteristics rather than a strong temporal or spatial bias.

Next, we look at 16 spatial modes extracted from wave clip 1 to examine the spatial patterns and variance explained visually.

The first 16 spatial modes, as seen in figure 4.6, reveal distinct structural patterns associated with the dynamics of the wave scene of the first wave clip. These modes are orthogonal, meaning each mode captures unique spatial information without overlapping with the other, forming a basis that allows any frame in the video to be reconstructed as a weighted combination of these spatial patterns and their corresponding temporal coefficients.

The first 16 principal components capture almost 50 percent of the total variance, indicating that a substantial portion of the dataset's variability is retained. This supports the use of these components for meaningful interpretation and analysis of the

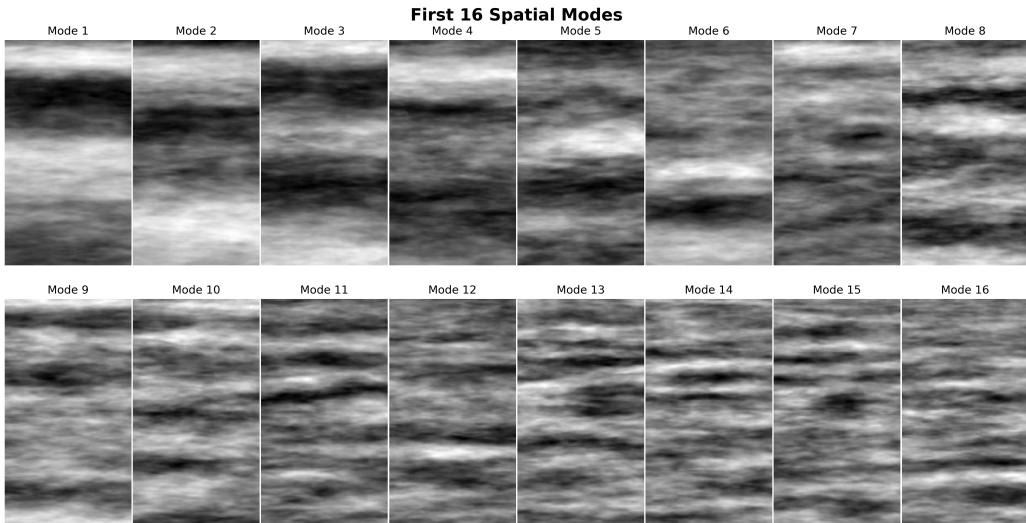


Figure 4.6: First 16 spatial modes extracted from Wave Clip 1 using low-rank decomposition. These modes highlight areas of strong variance and recurrent structural features in the wave motion, revealing coherent spatial dynamics across the color spectrum. The patterns help identify how each mode contributes to the observed motion over time.

underlying spatiotemporal patterns.

The first five spatial modes capture a large proportion of the dominant spatial variation present in this video(35%), likely capturing the propagation of waves across the surface with dominant horizontal gradients. These modes are responsible for capturing broad motion patterns and represent the largest-scale spatial. The next few sets of modes(PC6-PC9) have a similar horizontal structure but with a slight shift, likely showing secondary wave oscillations like back-and-forth ripples or surface reflections. Some spatial noise seems to appear, but it is not dominant. More localized features start to emerge at this point. The last few sets of modes are the least smooth of the set(PC9-PC16). They capture more fine-scale oscillatory patterns and small spatial variations. These higher modes act like corrections to the dominant pattern, adding finer texture to the frames.

Overall, these spatial modes serve as a low-dimensional basis capturing key structural features of the wave surface, capturing a substantial portion of the spatial structure of the wave clip. Next, to analyze the temporal patterns of this clip, we examine the first 25 temporal modes of this clip.

As can be inferred from figure 4.7, the first few modes (PC1-PC4) capture the most dominant wave patterns present in the clip. They capture the baseline motion of the wave clip. The graph shows smooth low-frequency oscillations, implying they capture dominant wave cycles and global intensity variations. The second set of temporal modes, ranging from PC5 through PC10, is still relatively smooth with some local bumps. These

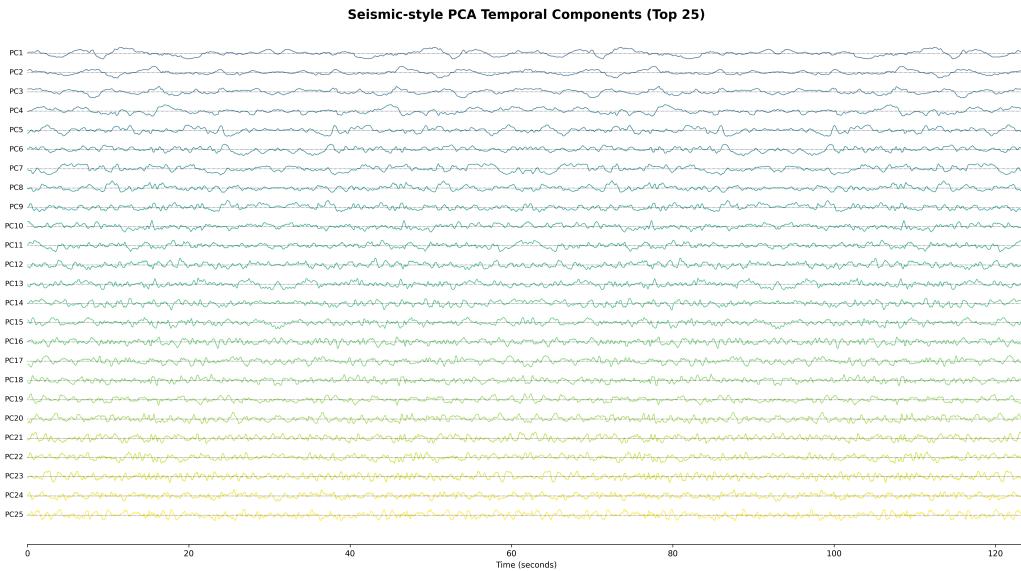


Figure 4.7: Plot of the first twenty-five temporal modes extracted from PCA of wave clip 1. The x-axis denotes the time in seconds from 0 to 120.

modes seem to be responsible for capturing secondary oscillations or regional variations in motion. The next few modes, ranging from PC11 through PC25, are high-frequency components but still retain some structure rather than pure noise. These modes reflect finer fluctuations while preserving temporal coherence.

Overall, these modes exhibit regular wave motion throughout the graph. The constant and regular nature of the graph also suggests a largely regular pattern, with occasional irregular events. This reconstruction highlights large-scale temporal patterns while also retaining meaningful higher-order structure.

Next, to examine how these temporal patterns manifest spatially, we observe the reconstructed frames at different time frames with different values of k . The original and reconstructed videos were plotted side by side and can be accessed by clicking the following link:

[Click Here to watch the Original Clip Plotted against Reconstructed Clips at different values of \$K\$](#)

These reconstructions were all obtained using the Dominant color Projection method as outlined previously

At $k=1$, most of the video is static as expected since the first spatial mode only captures the most dominant pattern in the video and hence results in the reconstructed video being a still image. As we keep increasing the number of modes, the reconstruction quality increases too; however, this comes with a tradeoff. While lower k values capture the core structure and dominant patterns, higher k values include finer details but may also begin to incorporate noise. After a certain point, increasing values of k result in almost diminishing results with only very fine changes barely noticeable, as



Figure 4.8: Comparison of the original and reconstructed frames from Wave Clip 1 at different time points, using different numbers of singular value components ($k = 1, 10, 50$). The first column shows the original frames, while subsequent columns display reconstructions with increasing k values.

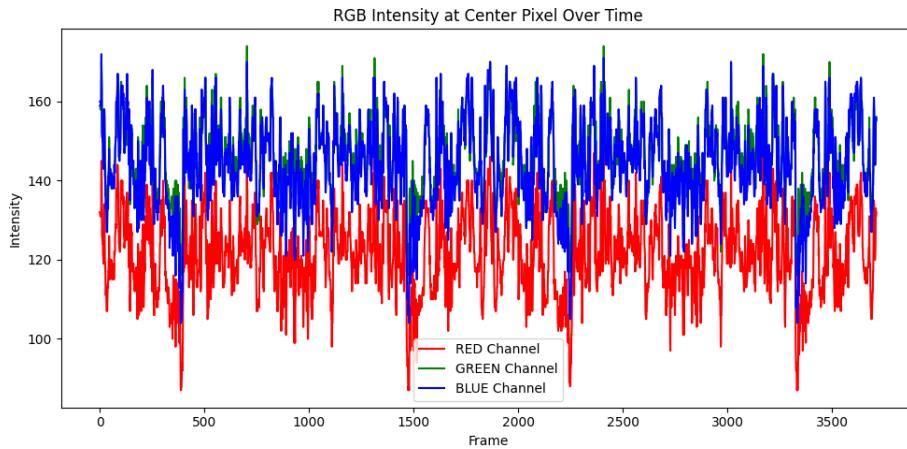


Figure 4.9: RGB intensity values of the center pixel over time in Wave Clip 1. The red, green, and blue channels exhibit fluctuating intensities that reflect local brightness changes due to wave motion. This signal offers insight into the temporal dynamics at a fixed spatial point and complements the global analysis captured by PCA.

can be seen in figure 4.8. Most of the dominant spatial patterns visible in the video are effectively captured at $k=10$. While increasing the number of modes beyond this point does improve the reconstruction fidelity, the improvements in the actual video quality become progressively smaller, resulting in diminishing returns. Additionally, higher values of k significantly increase the computational cost, often by an order of magnitude. At $k=50$, the reconstruction is highly similar to the original video with only minor differences noticeable, while at $k=100$, the reconstructed video is nearly indistinguishable from the original video, preserving both global and subtle wave dynamics

The original and reconstructed videos at different values of k can be accessed by clicking the link in the appendix. These side-by-side visualizations allow users to compare how well each k -value preserves essential wave dynamics. Next, we analyze the RGB intensity of the original clip at the center pixel to investigate color dynamics and assess the effectiveness of the dominant color projection method introduced earlier

The temporal RGB intensity plot of the center pixel over time, as seen in figure 4.9, reveals distinct color dynamics and patterns in the video that help analyze temporal variations, including lighting, wave oscillations, or natural dynamics. The center pixel was selected as a representative location to analyze color variation over time. The center pixel is often where dynamic changes occur in natural scenes, making it a useful point for observation and analysis.

As can be seen from the plot, the blue channel consistently reaches peak high values, indicating a strong emphasis on wave motion in the clip. This also suggests that the water does not absorb any shadows or sky reflection, with the emphasis being mostly on the water. The red channel, on the other hand, reaches the lowest values, suggesting a

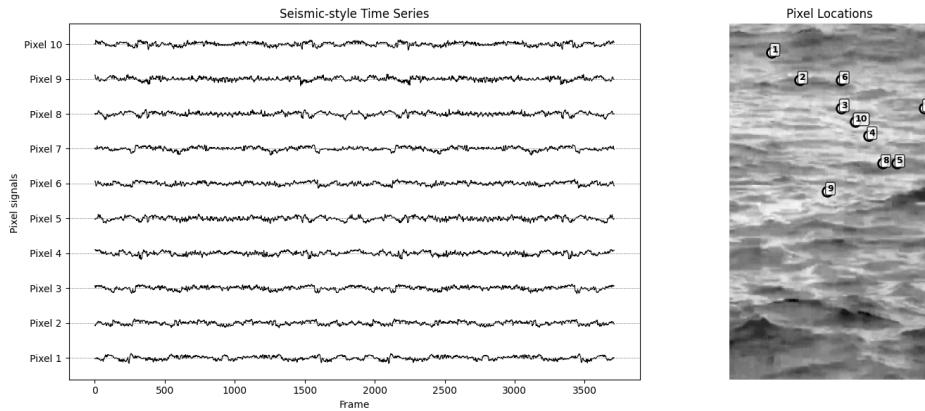


Figure 4.10: Time series Plot of ten random pixels from Wave clip 1

lack of sunlight reflections or bright patches where red is usually the strongest. It could also be an indication of no warm colored elements throughout the video. The green channel maintains consistent high values, coinciding with the blue channel, which could be indicative of either natural environments that often reflect green more, or the camera sensor's sensitivity to green light.

Overall, the RGB intensity analysis of the center pixel highlights that the dominant color components are primarily in the blue and green channels, while red remains minimal. These observations also validate the assumptions behind the dominant color projection method and provide a foundation for further spatiotemporal analysis based on the most influential color channels. Next, we extend our investigation to explore temporal relationships across multiple spatial locations by analyzing time series from ten representative pixels in the wave clip.

As we can infer from the figure 4.10, the signals are varying and relatively smooth with gradual fluctuations, implying the video progresses smoothly. While occasional jumps and spikes might correspond to sudden video changes such as a high wave tide, the overall temporal pattern of the clip is smooth and uniform. This behavior reflects simple spatial as well as temporal structure.

The preceding figures and analyses provide insight into the spatiotemporal structure of the wave clip. The next section summarizes these observations.

4.5 Wave Clip 2

This chapter examines the preprocessing and analysis of the first wave clip, which was captured in real-time during this study. The aim is to extract and analyze the dominant spatiotemporal pattern from this video using the methodology defined in the previous section and later compare its results with other video clips in the study.

4.5.1 Low Rank Decomposition on Wave Clip 2

We apply the previously discussed PCA formulation to extract dominant spatiotemporal patterns from large-scale wave clip 2. Reconstruction of the video was performed at different values of $k = 1, 10, 20, 50, 100$, and 200 . Due to computational limitations, we restricted our analysis to $k \leq 200$.

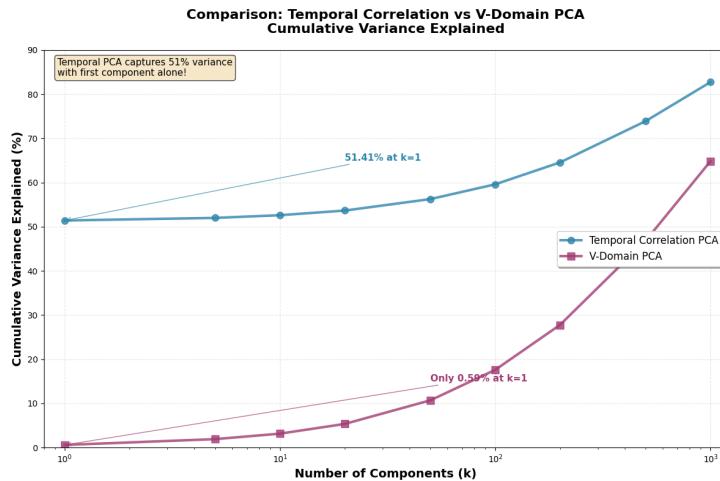


Figure 4.11: Cumulative variance explained by the top k principal components for Wave Clip 2

Table 4.3: Cumulative Variance Explained by Principal Components

Components (k)	Temporal Correlation PCA (%)	V-Domain PCA (%)
1	51.41	0.59
5	52.00	1.91
10	52.61	3.15
20	53.66	5.35
50	56.26	10.70
100	59.60	17.59
200	64.56	27.77
500	73.91	46.84
1000	82.74	64.83

Figure (fig 4.11) and table 4.3 showcase variance explained at different ranks on the correlation matrix of the temporal data and the projected data

Temporal correlation PCA is able to capture 51.41% of the variance explained. This implies that a dominant global temporal pattern is present throughout the video, with most frames participating in it. Spatially, only 0.59% of the variance was captured by

the first component, suggesting highly heterogeneous spatial patterns with no single spatial mode domination. We also notice how there is a slow buildup after the first component (51% at $k=1$ to 83% at $k=1000$), suggesting most of the temporal structure is captured by one major pattern. While looking at it spatially, it requires almost a thousand components to reach 65% variance, implying a very complex spatial structure with lots of independent spatial variations.

This analysis suggests that the wave has a highly predictable temporal structure, such as rhythmic waves and periodic motion, suggesting strong frame-to-frame correlation. The spatial structure, on the other hand, has many independent spatial regions that vary differently with a low spatial correlation. Next, we will inspect the first 16 spatial modes and analyze how the patterns captured by them visually appear.

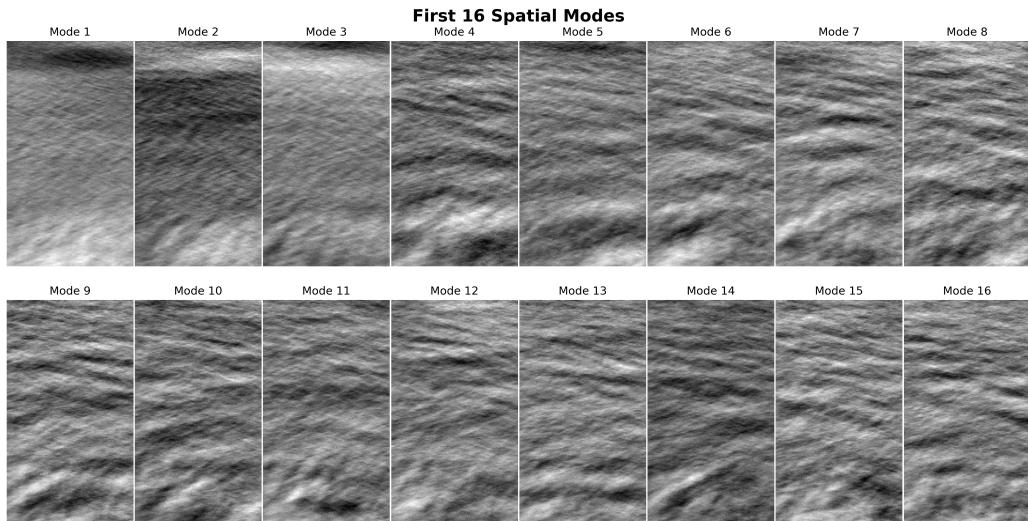


Figure 4.12: First 16 spatial modes extracted via PCA on the projected video data. These represent dominant spatial patterns across time, capturing coherent structures in wave motion. Mode 1 captures the most variance, followed by Modes 2 and 3.

Each spatial mode in figure 4.12 corresponds to a principal component, which captures a major pattern of spatial variation across all frames. Analyzing the spatial modes is an essential step in recognizing dominant spatial patterns in the wave clip.

The first few modes(PC1-PC4) capture smooth spatial fields and global wave structure, the most dominant spatial structure present across the video. The next few modes(PC5-PC8) capture orthogonal patterns that explain more subtle variations. These modes reveal spatial oscillations across the frame, such as ripples or intermediate wave crests. The last few modes(PC9-PC16) capture more densely patterned fields with rich local structure. This indicates the clip has complex spatial variability.

While each mode may not map directly to an interpretable feature in the physical world and does not indicate the presence of a single dominant spatial mode, collectively

they form the orthogonal spatial basis used for frame reconstruction. Their visual structure suggests they capture the evolving contrast and waveforms in the scene, even if the precise semantics are abstract. Next, we analyze the temporal patterns of this clip and interpret its temporal dynamics.

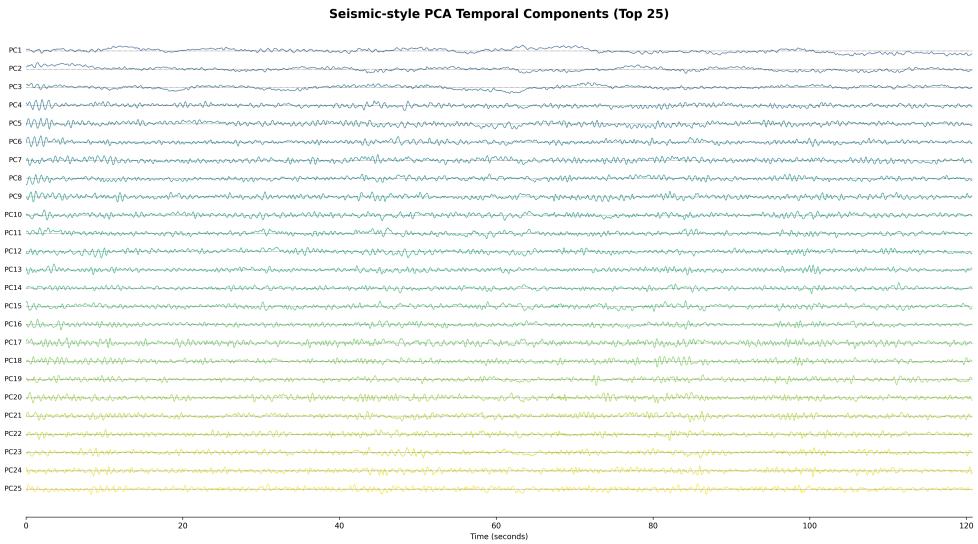


Figure 4.13: Temporal PCA modes for the first twenty-five spatial components of the projected wave video. Mode 1 remains dominant throughout most of the frames before tapering off.

By observing figure 4.13, we can find and interpret dominant temporal patterns from wave clip 2.

The first three temporal modes seem to capture broad oscillatory motion, representing the dominant wave frequencies and fundamental modes. These modes contain the most significant part of the wave's energy. The second set of temporal modes, ranging from PC4 through PC10, shows finer oscillations that could be harmonic frequencies. These are the secondary modes of the wave data. The next few temporal modes (PC9-PC15) display increasing complexity, capturing dispersion effects and nonlinear wave interactions. These are the higher-order modes of the clip. The final modes from PC16 to PC25 represent the finer structures of the clip, representing small-scale turbulence or very high-frequency wave components. Overall, the clear hierarchy of these modes suggests the existence of well-defined dominant modes with decreasing contributions in energy. Due to the high spatial complexity of the video, we also analyze the mean and standard deviation images of the clip to understand how much data is distributed across frames.

As can be seen from figure 4.14, the mean image is almost entirely black, with only a faint white corner indicating that most pixels fluctuate around zero and no single region dominates the scene. In contrast, the standard deviation Image has more structure

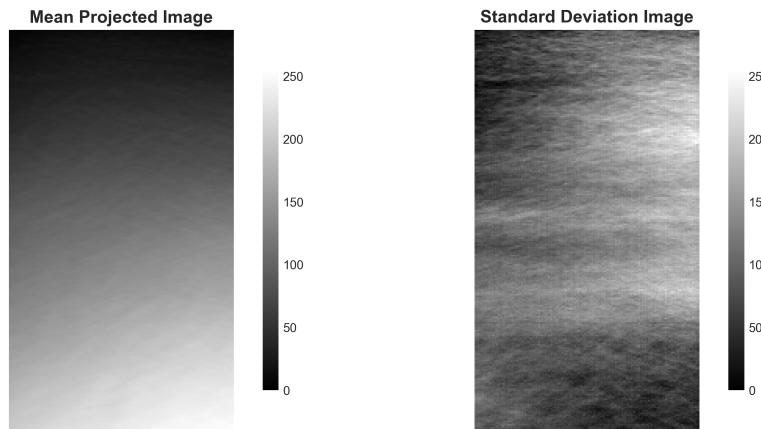


Figure 4.14: Analysis of a sample frame from Wave Clip 2. left: mean projected image; right: Standard Deviation Image. This illustrates the spatial variation captured by the dominant color projection.

with faint wave lines, showing where motion is coherent. This also highlights how most pixels experience significant temporal variation.

To gain a better understanding of how these reconstructed frames differ from the original video, we reconstruct the video up to 200 components. The original and reconstructed videos were plotted side by side and can be accessed by clicking the following link:

[Click Here to watch the Original Clip Plotted against Reconstructed Clips at different values of \$K\$](#)

As can be seen from Figure 4.15, the reconstruction at $k=5$ only captures the most dominant features and doesn't accurately capture the visual fidelity and finer details of the original. At $k = 50$, we start to notice the finer details, and the wave motions begin to look a lot more similar to the original. As we further increase k , the reconstruction retains increasingly more visual detail. By $k = 200$, many of the finer details are captured, and the frame starts to look a lot more similar to the original. To capture all the variance explained and retain visual fidelity similar to the original, we will have to keep increasing the value of k ; however, increasing k comes with trade-offs. Higher values of k require significantly more computational power, making it hard and expensive to execute on a standard machine. As we increase the number of PCs, we might lose the advantage of compression, which is one of the key benefits of PCA. Visual differences in higher grades, such as $k = 400$ and $k = 500$, may almost be imperceptible to the human eye, which is a waste of resources for a simple improvement of 2% explained variance.

It is best to find an optimal value of k , which involves balancing reconstruction quality, computational efficiency, and compression. If you aim to capture near lossless reproduction, reconstruction at higher values of k may be worth looking into, for exploratory analysis or pattern detections; lower values of k often suffice.

Figure 4.16 shows the temporal intensity of the center pixel across the three RGB

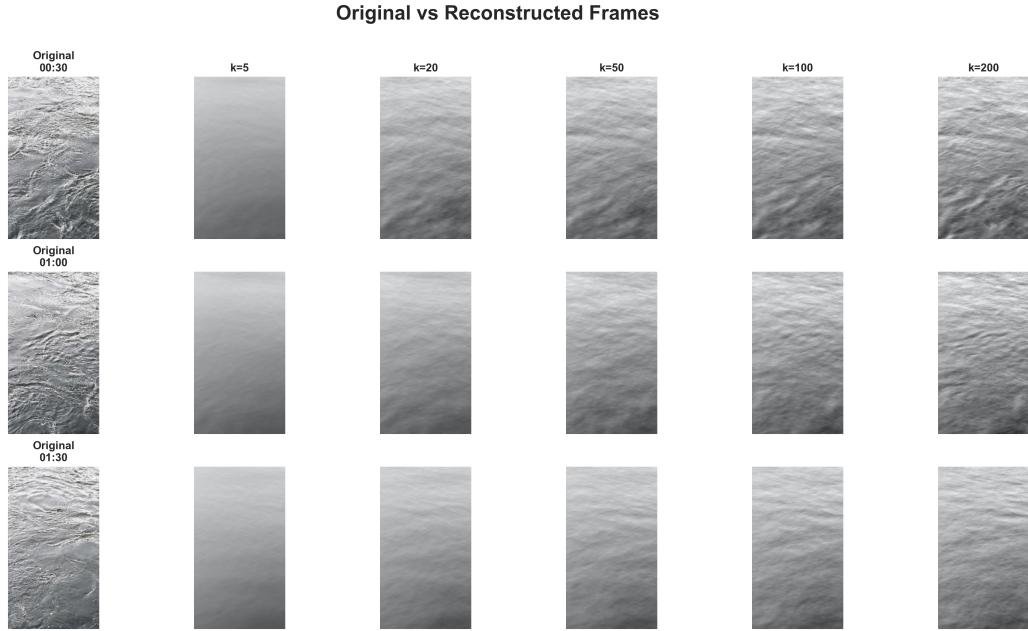


Figure 4.15: Comparison of original vs reconstructed frames at $k = 5, 20, 50, 100, 200$ across different frames in the sequence. As k increases, the reconstruction captures details and approaches the original in visual fidelity.

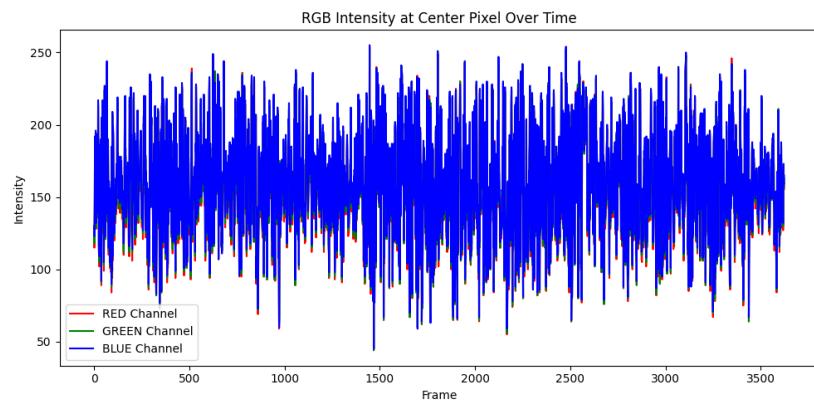


Figure 4.16: Center pixel RGB intensity over time for Wave Clip 2. This plot illustrates the dominant color channel, supporting the choice of projecting the video data onto the dominant color for PCA

channels. The blue channel clearly dominates throughout the scene, remaining consistently higher than both the red and green channels, which are comparatively almost negligible. This observation aligns with the visual appearance of the dataset and confirms projecting the video onto the dominant color axis. By doing so, we also avoid redundancy from a weaker channel and reduce complexity, ensuring PCA is applied to the component carrying the most meaningful variation. However, it is also worth mentioning that this approach discards subtle variations present in the red and green channels, which could contain secondary information.

Establishing that the blue channel dominated the main temporal variation, we now examine how these temporal fluctuations are distributed across multiple pixels. To do this, we analyze the time series plot of 10 random pixels from wave clip 2.

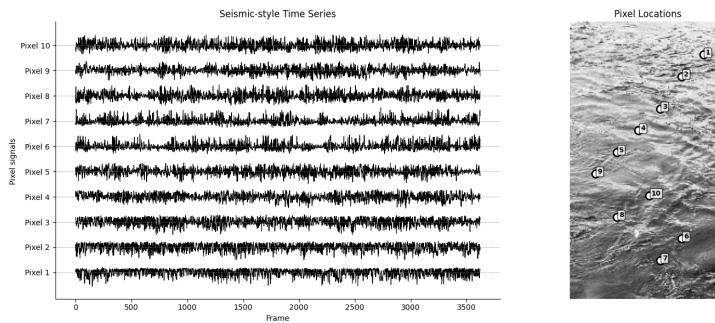


Figure 4.17: Time series Plot of ten random pixels from Wave Clip 2.

As we can observe from figure 4.17, these fluctuations translate into a higher proportion of variance explained in the temporal domain, which matches with our results obtained previously. The associated temporal components with these pixels capture a set of independent patterns. This aligns with the fact that the sampled region of the video is spatially more complex. This analysis also reiterates the fact that the wave clip is temporally more dominant.

4.6 Summary of Key Observations for Large-Scale Wave Clips

Having thoroughly analyzed Wave Clips 1 and 2 through dimensionality reduction techniques, including PCA-based reconstruction, spatial and temporal mode decomposition, and variance explained metrics, we have established a clear understanding of the dominant patterns and their temporal dynamics within the video data. The reconstruction comparisons demonstrated the trade-offs between compression and visual fidelity, while the spatial and temporal modes provided insight into the underlying wave structures and their evolution over time. With these results in hand, we now turn to a detailed discussion of the implications, limitations, and potential applications of this methodology, along with suggestions for future work.

Chapter 5

Discussion

This chapter synthesizes the findings from the spatiotemporal analysis of wave video data using PCA and SVD. By comparing results across different clips and reconstruction ranks, we evaluate the strengths and limitations of the proposed methodology. The discussion also reflects on the insights gained from both small-scale and large-scale examples and explores implications for future applications to similar datasets.

5.1 Overview of Quantitative Video Analysis

This section discusses the novelty of extracting quantitative information from spatiotemporal data, particularly in the context of wave video analysis. The methodology demonstrates that even nontraditional sources of dataâsuch as videosâcan be meaningfully explored using dimensionality reduction techniques like PCA and SVD to yield interpretable temporal and spatial patterns.

Similar methodologies have been applied in other domains. For instance, Su et al. [Su et al. 2014] analyzed complex temporal patterns on Riemannian manifolds, with applications ranging from bird migration and hurricane tracking to video surveillance. Laptev [Laptev 2005] proposed the extraction of space-time interest points from video data, laying the groundwork for quantitative feature extraction from dynamic scenes.

Traditionally, video data is consumed qualitatively; although visual inspection may help identify meaningful events, it fails to capture subtle or repetitive dynamics, especially when motion is minimal. In contrast, Quantitative analysis can uncover patterns and dynamics not easily perceived by the human eye. This is especially valuable in scientific and environmental domainsâsuch as ocean wave monitoring or meteorological footageâwhere interpreting motion patterns visually is often challenging.

5.1.1 Comparison of Results between Small-Scale and Large-Scale Video Datasets

In this subsection, we compare the outcomes of applying PCA-based spatiotemporal analysis on two distinct video datasets with equally different characteristics, differing in scale and complexity. The small-scale dataset, on the other hand, is characterized by lower resolution and shorter duration, serves as a controlled example to validate the methodology.

By examining the differences between the spatiotemporal patterns across these datasets, we can assess the scalability and usefulness of PCA on spatiotemporal datasets.

Table 5.1: Comparison of Key Dataset Characteristics

Dataset	Resolution (pixels)	Duration (min:sec)	Notes on Complexity
Small-scale clip	140 × 154	0:59	Low resolution, short duration, Moderate complexity
Wave Clip 1	1920 × 1080	2:03	High resolution, longer duration, moderate complexity
Wave Clip 2	1920 × 1080	2:00	High resolution, longer duration, higher complexity, more variance

As we can infer from Table 5.2 Wave Clip 1 captures variance more quickly compared to Wave Clip 2. By $k=100$, over 84% of the variance is explained in both the temporal correlation and projected domain. In contrast, Wave clip 2 explains 60% of the temporal variance by $k=100$ and reaches only 83% at $k=1000$.

This showcases the high spatial complexity of wave clip 2 in comparison to wave clip 1. This also represents the presence of a dominant temporal mode in wave clip 2 as compared to wave clip 1.

These results suggest that while PCA is highly effective for less complex datasets, it becomes increasingly more difficult for PCA to capture patterns effectively as variability increases.

Table 5.2: Comparison of Cumulative Variance Explained by Principal Components for Wave Clip 1 and Wave Clip 2

Components (k)	Wave Clip 1 (%)		Wave Clip 2 (%)	
	Temporal Corr. PCA	V-Domain PCA	Temporal Corr. PCA	V-Domain PCA
1	10.30	10.16	51.41	0.59
5	35.36	36.37	52.00	1.91
10	47.56	49.28	52.61	3.15
20	57.94	59.44	53.66	5.35
50	73.00	73.88	56.26	10.70
100	84.02	84.49	59.60	17.59
200	92.30	92.52	64.56	27.77
500	97.26	97.36	73.91	46.84
1000	98.68	98.73	82.74	64.83

Comparing these results with the variance explained results of the small-scale wave clip (see table 4.1), we observe how Wave clip 1, even after having higher dimensionality (1920x1080), ends up capturing more variance in comparison to the small-scale clip. This shows how dimensionality doesn't play the biggest role when extracting spatiotemporal patterns from video clips if the spatial and temporal complexity of the video is high. The same can be said about wave clip 2, where PCA can easily capture the most dominant temporal mode even after high spatial but lower temporal complexity. This suggests that PCA is effective no matter the dimensionality of the dataset.

To better capture the trade-off between reconstruction fidelity and dimensionality, we examine the reconstruction percentage errors against the number of components.

As can be seen from figure 5.1, the steep decline in MSE highlights that the first few principal components capture the bulk of the meaningful variance in the first wave clip. At higher values of k, such as 200, the MSE falls to almost 8 %, approaching a near complete reconstruction with diminishing results beyond this point. Similar comparisons were done for all other datasets.

Since PCA is a linear method, it suffers when dealing with heterogeneous spatiotemporal data. Its utility diminishes, suggesting the use of more advanced nonlinear or deep learning based methods when dealing with such intrinsic data. These findings underscore challenges related to scalability and computational cost when extending PCA-based methodologies to large-scale video data, highlighting the need to balance fidelity with efficiency in real-world applications.

5.2 Effectiveness of PCA in spatiotemporal Video Analysis

This section aims to identify the effectiveness of PCA in spatiotemporal data analysis, focusing mainly on video data.

By looking at our experiments conducted above, as seen in table 5.2, we can see how PCA is highly effective as a linear tool in compressing data and retaining important

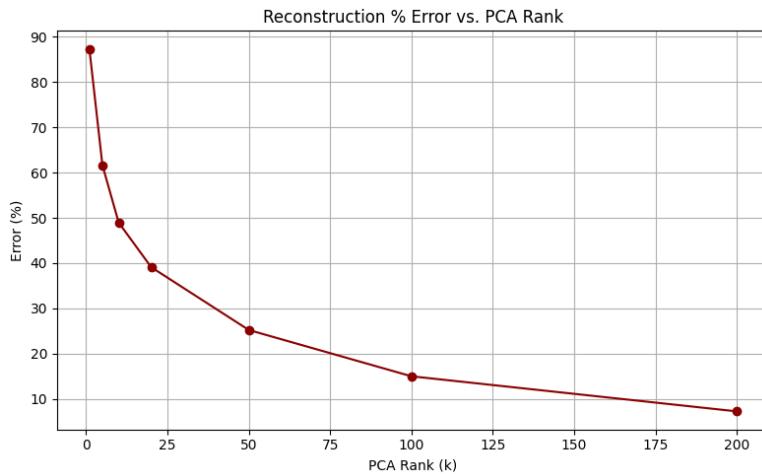


Figure 5.1: Reconstruction error of the projected video as a function of PCA rank k . The mean squared error decreases with increasing rank, showing how the low-rank approximation captures the dominant spatial-temporal patterns. The right axis can optionally indicate the percentage error relative to the total variance.

spatial and temporal patterns, especially in smaller datasets with low to moderate complexity. PCA is the most effective and best way to quantitatively analyze such data linearly. We can also refer to the spatial and temporal modes visualizations for all three datasets and observe how PCA effectively captures dominant spatial and temporal modes from such datasets and can capture important patterns by analyzing these graphs and figures.

PCA serves as a powerful tool for quantitative analysis of spatiotemporal data, although its linearity might hold it back when dealing with complex patterns or nonlinear dynamics. It is still a fairly robust and efficient method.

5.3 Relation to Climatology and Environmental Monitoring

Principal Component Analysis (PCA) and its variants have been widely used in climatology to identify dominant spatiotemporal modes in environmental data. For instance, *Rotated Spectral Principal Component Analysis* (rsPCA) has been applied to detect dynamical modes of variability in climate systems, particularly focusing on sea surface temperature patterns [Clement, and et al. 2018]. Similarly, *Flow-directed PCA* has been utilized for analyzing monitoring networks to identify common spatiotemporal patterns in flow-connected sites [Gallacher et al. 2015]. These studies demonstrate the efficiency of PCA-based methods in extracting meaningful patterns from complex,

high-dimensional environmental datasets, facilitating improved understanding and monitoring of climatological processes. Our application of PCA to wave surface video data aligns with these approaches, providing a novel way to capture coherent spatiotemporal variability relevant to environmental monitoring and climatological studies.

Wave surface data represents a crucial component of coastal and oceanographic studies. Changes in wave height, frequency, and patterns are directly linked to broader climate dynamics, making their analysis vital for environmental monitoring and forecasting. While numerous studies have focused on these wave characteristics and their influence on climate patterns, relatively fewer have concentrated on extracting quantitative spatiotemporal information directly from wave surface data to enable detailed analysis and modeling, which is what this study has aimed to do.

5.4 Applications and Broader Implications

PCA-based spatiotemporal analysis is an essential multivariate analysis method. This methodology described in the study, can have use cases in environmental monitoring, which can aid in the detection of climate-driven changes in ocean waves, flood prediction, and coastal erosion.

Previous studies, such as North et al. [1982], have analyzed the sampling errors in the eigenvectors of covariance matrices with applications to atmospheric data, using PCA to investigate parameters of interest in climate change. Another important reference is the work by Wilks, which provides a comprehensive review of statistical methods in the atmospheric sciences [Wilks 2011], analyzing the usefulness of these methods in climate model evaluation and related applications. These studies show the effectiveness of methods such as PCA and its broader implications outside of wave surface analysis.

The methodology and analysis provided in this study could serve as a basis for other disciplines in spatiotemporal analysis, such as video compression and pattern recognition in datasets such as weather radars and satellite imagery. By providing a quantitative approach to capture dominant spatial and temporal structures, this approach offers a valuable tool for advancing research and practical applications in various scientific and engineering fields.

5.5 Limitations of the Current Study

Despite the promising results demonstrated in this study, it is important to acknowledge the limitations present in this study. Since PCA follows Linear Dimensionality Reduction methodology, it is not able to capture patterns effectively when dealing with nonlinear patterns with a high level of complexity(refer to table 5.2). To deal with such data, it is important to explore nonlinear methods like Kernel PCA, or autoencoders, and pair them with PCA to better capture complex dynamics.

This study has only focused on a specific kind of spatiotemporal dataset. Analyzing larger and more diverse datasets is important to assess the generalizability of the

methodology across different environments. Additionally, dealing with more diverse and larger datasets requires high computational power, which cannot be achieved on standard machines. Dealing with such datasets might require optimizing computations and using GPU acceleration to make it feasible to handle, which is costly to access. Larger real-world datasets might also contain much more noise, such as lighting changes at each frame and camera motion, which PCA cannot explicitly account for. Finally, due to high computational cost and limited resources, down-sampling was done, which may have affected the generalizability of PCA.

5.6 Future Work

Future work could involve developing a user-friendly application or software tool to automate the spatiotemporal analysis of wave surface videos, making the methodology more accessible to researchers and practitioners in environmental monitoring. Further exploration of nonlinear dimensionality reduction techniques and real-time processing capabilities would also enhance the applicability of this approach.

Chapter 6

Conclusion

This study applied PCA/SVD to distinct wave video datasets to analyze spatiotemporal patterns in wave video data. This study presented a quantitative analysis of spatiotemporal video datasets and interpreted meaningful analyses from it. In less complex datasets, relatively more variance was explained, while datasets with higher complexity required relatively higher components to retain the full variance. Each video had distinct features and spatial and temporal patterns, making it the perfect set of datasets to run these analyses on. This study explored how, while PCA can extract dominant patterns effectively, no matter the dimensionality of the data, the linearity of PCA came across as a limitation to reconstruct videos with higher spatial or temporal complexity and required a massive number of modes, resulting in a really high computational load, making it almost non-feasible on normal workstations. This approach, outlined in this study, provides a robust framework for quantitative analysis of video-based spatiotemporal data, with potential applications in environmental monitoring, wave forecasting, pattern recognition, and analysis of different types of spatiotemporal datasets.

Appendix A

Key Code Blocks

A.1 PCA-Based RGB Channel Decomposition Code for Decomposition and Reconstruction for small wave clip

```
library(av)
library(magick)
library(png)
library(ggplot2)

read_frames <- function(video_path) {
  temp_dir <- tempfile("frames")
  dir.create(temp_dir)
  av::av_video_images(video_path, destdir = temp_dir, format = "png")
  frame_files <- list.files(temp_dir, full.names = TRUE, pattern = "\\\\.png$")
  lapply(frame_files, image_read)
}

flatten_frame <- function(img) {
  data <- image_data(img, channels = "rgb")
  r <- as.numeric(data[1, , ])
  g <- as.numeric(data[2, , ])
  b <- as.numeric(data[3, , ])

reconstruct_frame <- function(vec, height, width) {
  n_pix <- height * width
```

```

r_vec <- vec[1:n_pix]
g_vec <- vec[(n_pix + 1):(2 * n_pix)]
b_vec <- vec[(2 * n_pix + 1):(3 * n_pix)]

gamma <- 1.8
r_vec <- ((pmax(0, pmin(255, r_vec)) / 255) ^ (1/gamma)) * 255
g_vec <- ((pmax(0, pmin(255, g_vec)) / 255) ^ (1/gamma)) * 255
b_vec <- ((pmax(0, pmin(255, b_vec)) / 255) ^ (1/gamma)) * 255

r_vec <- round(r_vec)
g_vec <- round(g_vec)
b_vec <- round(b_vec)

r_mat <- matrix(r_vec, nrow = width, ncol = height)
g_mat <- matrix(g_vec, nrow = width, ncol = height)
b_mat <- matrix(b_vec, nrow = width, ncol = height)

temp_file <- tempfile(fileext = ".png")

img_array <- array(0, dim = c(height, width, 3))
img_array[, , 1] <- t(r_mat) / 255 # PNG expects values 0-1, not 0-255
img_array[, , 2] <- t(g_mat) / 255
img_array[, , 3] <- t(b_mat) / 255

png::writePNG(img_array, temp_file)
image_read(temp_file)
}

# Flatten all frames
cat("Flattening all frames for PCA...\n")
data_mat <- t(do.call(cbind, lapply(imgs, flatten_frame))) # frames x (pixels*3)
cat("Data matrix dimensions:", dim(data_mat), "\n")

# Perform PCA
cat("Performing PCA...\n")

```

```
pca_res <- prcomp(data_mat, center = TRUE, scale. = FALSE)

reconstruct_video <- function(k) {
  cat(sprintf("==> RECONSTRUCTING WITH k=%d COMPONENTS ==>\n", k))

  k_actual <- min(k, ncol(pca_res$x))
  if (k_actual != k) {
    cat("Warning: Reduced k from", k, "to", k_actual, "(max available)\n")
  }

  # PCA reconstruction
  scores <- pca_res$x[, 1:k_actual, drop = FALSE]
  loadings <- pca_res$rotation[, 1:k_actual, drop = FALSE]

  recon <- scores %*% t(loadings)
  recon <- sweep(recon, 2, pca_res$center, "+")

  output_dir_k <- file.path(output_dir, sprintf("frames_k%03d", k))
  dir.create(output_dir_k, showWarnings = FALSE)

  cat("Reconstructing", n_frames, "frames...\n")
  start_time <- Sys.time()

  for (i in 1:n_frames) {
    frame_vec <- recon[i, ]
    frame_img <- reconstruct_frame(frame_vec, height, width)

    output_path <- file.path(output_dir_k, sprintf("frame_%04d.png", i))
    image_write(frame_img, path = output_path)

    if (i %% 20 == 0 || i == n_frames) {
      elapsed <- as.numeric(Sys.time() - start_time, units = "secs")
      cat(sprintf("  Frame %d/%d (%.1f%%) - %.1fs elapsed\n",
                 i, n_frames, 100*i/n_frames, elapsed))
    }
  }

  cat("Encoding video...\n")
  png_files <- list.files(output_dir_k, pattern = "\\.png$", full.names = TRUE)
```

```
png_files <- sort(png_files) # Ensure correct order

output_video <- file.path(output_dir, sprintf("reconstructed_k%03d.mp4", k))
av_encode_video(png_files, output = output_video, framerate = fps)

var_pct <- ifelse(k_actual <= length(cumvar), cumvar[k_actual], 100)

cat(sprintf(" Video saved: %s\n", basename(output_video)))
cat(sprintf(" Variance explained: %.2f%%\n", var_pct))
cat(sprintf(" Time taken: %.1f seconds\n\n",
            as.numeric(Sys.time() - start_time, units = "secs")))

return(list(k = k_actual, variance_explained = var_pct))
}
```

A.2 PCA Analysis on Projected Grayscale Video Derived from RGB Dominant Color Basis in Python

```
import cv2
import numpy as np
import os
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
video_path = os.path.expanduser("~/Downloads/wave1_clip_final_1080p.mp4")
output_dir = os.path.expanduser("~/Downloads/wave_output")
os.makedirs(output_dir, exist_ok=True)
resize_dim = (256, 256)

def read_video(path, max_frames=None, resize_dim=None):
    cap = cv2.VideoCapture(path)
    frames = []
    count = 0
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret or (max_frames and count >= max_frames):
            break
        if resize_dim:
            frame = cv2.resize(frame, resize_dim)
        frames.append(frame)
    cap.release()
    return np.array(frames)
```

```

if resize_dim is not None:
    frame = cv2.resize(frame, resize_dim)
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frames.append(rgb_frame)
count += 1
cap.release()
return np.array(frames)

def save_video(frames_rgb, output_path, fps):
    h, w, _ = frames_rgb[0].shape
    out = cv2.VideoWriter(output_path, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
    for frame in frames_rgb:
        bgr = cv2.cvtColor(frame.astype(np.uint8), cv2.COLOR_RGB2BGR)
        out.write(bgr)
    out.release()

video = read_video(video_path, resize_dim=resize_dim)
T, H, W, C = video.shape
print("Loaded video:", video.shape)

video_flat = video.reshape(-1, 3)
mean_rgb = video_flat.mean(axis=0)
std_rgb = video_flat.std(axis=0)
normalized = (video_flat - mean_rgb) / std_rgb
corr = np.corrcoef(normalized.T)
eigvals, eigvecs = np.linalg.eig(corr)
top3_vecs = eigvecs[:, np.argsort(eigvals)[-3:]] # (3, 3)
dominant_vec = top3_vecs[:, -1] # single dominant direction

projected = np.empty((T, H, W))
for t in range(T):
    frame_flat = video[t].reshape(-1, 3)
    projection = (frame_flat - mean_rgb) @ dominant_vec # centered projection
    projected[t] = projection.reshape(H, W)

mean_image = projected.mean(axis=0)
std_image = projected.std(axis=0)

```

```
standardized = (projected - mean_image) / std_image

X = standardized.reshape(T, -1)
pca = PCA(n_components=20)
X_pca = pca.fit_transform(X)
print("Explained variance ratio (top 5):", pca.explained_variance_ratio_[:5])

fps = get_video_fps(video_path)
k_values = [1, 5, 10, 20]

for k in k_values:
    pca_k = PCA(n_components=k)
    X_k = pca_k.fit_transform(X)
    Xk_reconstructed = pca_k.inverse_transform(X_k)
    reconstructed = Xk_reconstructed.reshape(T, H, W) * std_image + mean_image

    # Back-project to RGB using dominant direction + mean RGB
    rgb_recons = []
    for t in range(T):
        flat = reconstructed[t].flatten()
        rgb_flat = (flat[:, None] * dominant_vec[None, :]) + mean_rgb # shape: (H*W, 3)
        rgb_frame = rgb_flat.reshape(H, W, 3)
        rgb_frame = np.clip(rgb_frame, 0, 255)
        rgb_recons.append(rgb_frame.astype(np.uint8))
```

Appendix B

Additional Resources

This appendix contains useful links related to the project. All links are clickable.

B.1 Code and Data Repository

The code for all analyses and visualizations of the three wave clips is available in the project repository on GitHub:

[Github Repository with Code](#).

All datasets used in this project can be accessed by clicking the links below:

- [Small Scale Wave Clip](#)
- [Wave Clip 1](#)
- [Wave Clip 2](#)
- [All Clips and Reconstructions](#)

B.2 Video Demonstrations

All Small Scale Clip Reconstructions can be accessed by clicking the links below :

- [Small Scale Wave Clip Reconstructed at \$k = 1\$](#)
- [Small Scale Wave Clip Reconstructed at \$k = 5\$](#)
- [Small Scale Wave Clip Reconstructed at \$k = 10\$](#)
- [Small Scale Wave Clip Reconstructed at \$k = 20\$](#)
- [Small Scale Wave Clip Reconstructed at \$k = 50\$](#)
- [Small Scale Wave Clip Reconstructed at \$k = 100\$](#)

All Wave Clip 1 Reconstructions can be accessed by clicking the links below:

- [Wave Clip 1 Reconstructed at \$k = 1\$](#)
- [Wave Clip 1 Reconstructed at \$k = 5\$](#)
- [Wave Clip 1 Reconstructed at \$k = 10\$](#)
- [Wave Clip 1 Reconstructed at \$k = 20\$](#)
- [Wave Clip 1 Reconstructed at \$k = 50\$](#)
- [Wave Clip 1 Reconstructed at \$k = 100\$](#)
- [Wave Clip 1 Reconstructed at \$k = 200\$](#)

All Wave Clip 2 Reconstructions can be accessed by clicking the links below:

- [Wave clip 2 Reconstructed at \$k = 5\$](#)
- [Wave clip 2 Reconstructed at \$k = 20\$](#)
- [Wave clip 2 Reconstructed at \$k = 50\$](#)
- [Wave clip 2 Reconstructed at \$k = 100\$](#)
- [Wave clip 2 Reconstructed at \$k = 200\$](#)

Bibliography

- Almar, Rafael et al. [2022]. “Global Satellite-Based Coastal Bathymetry from Waves”. In: Accessed: 2025-08-13. URL: https://horizon.documentation.ird.fr/exl-doc/pleins_textes/2022-01/010083830.pdf.
- Almeida, J., and R. Almar [2020]. “Application of Remote Sensing Methods to Monitor Coastal Zones”. In: *Journal of Marine Science and Engineering* 8.6, p. 391. DOI: [10.3390/jmse8060391](https://doi.org/10.3390/jmse8060391). URL: <https://www.mdpi.com/2077-1312/8/6/391>.
- Bilinski, Piotr, and Francois Bremond [2015]. “Video Covariance Matrix Logarithm for Human Action Recognition in Videos”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI/AAAI Press, pp. 2140–2147. DOI: [10.5555/2832415.2832675](https://doi.org/10.5555/2832415.2832675). URL: <https://doi.org/10.5555/2832415.2832675>.
- Brown, Jesse et al. [2022]. “A dynamic gradient architecture generates brain activity states”. In: *NeuroImage* 261, p. 119526. DOI: [10.1016/j.neuroimage.2022.119526](https://doi.org/10.1016/j.neuroimage.2022.119526). URL: https://www.researchgate.net/publication/362360326_A_dynamic_gradient_architecture_generates_brain_activity_states.
- Bueso, Diego, Maria Piles, and Gustau Camps-Valls [2020]. “Nonlinear PCA for Spatio-Temporal Analysis of Earth Observation Data”. In: *arXiv preprint arXiv:2002.04539*. URL: <https://arxiv.org/abs/2002.04539>.
- Caffo, Brian S et al. [2010]. “Two-stage decompositions for the analysis of functional connectivity for fMRI with application to Alzheimer’s disease risk”. In: *NeuroImage* 51.3, pp. 1140–1149. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2862769>.
- Camus, Paula et al. [2013]. “High resolution downscaled ocean waves (DOW) reanalysis in coastal areas”. In: *Coastal Engineering* 72, pp. 56–68. DOI: [10.1016/j.coastaleng.2012.09.002](https://doi.org/10.1016/j.coastaleng.2012.09.002).
- Casella, Elisa et al. [2020]. “Accuracy of sand beach topography surveying by drones and photogrammetry”. In: *Geo-Marine Letters* 40.2, pp. 255–268. DOI: [10.1007/s00367-020-00638-8](https://doi.org/10.1007/s00367-020-00638-8). URL: <https://link.springer.com/article/10.1007/s00367-020-00638-8>.

- Castelle, Bruno et al. [2025]. "Semi-empirical forecast modelling of rip-current and shore-break wave hazards". In: *Natural Hazards and Earth System Sciences* 25.9, pp. 2379–2394. DOI: 10.5194/nhess-25-2379-2025.
- Cavallo, Andrea, Mohammad Sabbaqi, and Elvin Isufi [2024]. "Spatiotemporal Covariance Neural Networks". In: *arXiv preprint arXiv:2409.10068*. URL: <https://arxiv.org/abs/2409.10068>.
- Chen, Jinhe [2018]. "Spectrum Features Analysis of Ocean Wind and Wave Based on Fast Fourier Transform Algorithm". In: *Journal of Coastal Research* 83.10083, pp. 418–422. DOI: 10.2112/SI83-070.1.
- Chen, Xinyu et al. [2020]. "Scalable Low-Rank Tensor Learning for Spatiotemporal Traffic Data Imputation". In: *arXiv preprint arXiv:2008.03194*. URL: <https://arxiv.org/abs/2008.03194>.
- Christiansen, Bo [2005]. "The Shortcomings of Nonlinear Principal Component Analysis in Identifying Circulation Regimes". In: *Journal of Climate* 18.22, pp. 4814–4823. DOI: 10.1175/JCLI3569.1.
- Clement, A. C., and et al. [2018]. "Rotated Spectral Principal Component Analysis for Identifying Dynamical Modes of Variability in Climate Systems". In: *Journal of Climate* 31, pp. 1234–1250.
- Dean, Robert G, and Robert A Dalrymple [2004]. *Coastal Processes With Engineering Applications*. Cambridge University Press.
- Ding, Yuanming et al. [2025]. "Traffic flow prediction based on spatiotemporal encoder-decoder model". In: *PLOS ONE* 20.5, e0321858. DOI: 10.1371/journal.pone.0321858. URL: <https://doi.org/10.1371/journal.pone.0321858>.
- Duan, Yuting, Xiaotong Li, and Liyang Zhang [2024]. "Marine Surface Wave Recognition and Its Application in Safety and Forecasting". In: *Ocean Engineering* 285, p. 116372.
- Gallacher, K. et al. [2015]. "Flow-directed Principal Component Analysis for Monitoring Networks". In: *Environmental Modelling Software* 72, pp. 35–46.
- Gong, Wenwu, Zhejun Huang, and Lili Yang [2023]. "Spatiotemporal Regularized Tucker Decomposition Approach for Traffic Data Imputation". In: *arXiv preprint arXiv:2305.06563*. URL: <https://arxiv.org/abs/2305.06563>.
- Gyimadu, Michael, and George Bell [2025]. *A Comparative Analysis of Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) as Dimensionality Reduction Techniques*. DOI: 10.48550/arXiv.2506.16663. arXiv: 2506.16663 [cs.LG]. URL: <https://arxiv.org/abs/2506.16663>.
- He, Chen et al. [2025]. "Tensor dynamic mode decomposition for multidimensional data". In: *arXiv preprint arXiv:2508.02627*. URL: <https://arxiv.org/abs/2508.02627>.
- Holman, Rob, Nathaniel Plant, and Todd Holland [2013]. "cBathy: A robust algorithm for estimating nearshore bathymetry from video imagery". In:

- Coastal Engineering* 81, pp. 118–129. DOI: 10.1016/j.coastaleng.2013.07.003.
- Jang, Jun-Gi et al. [2018]. *Zoom-SVD: Fast and Memory Efficient Method for Extracting Key Patterns in an Arbitrary Time Range*. arXiv: 1805.00754 [cs.DB]. URL: <https://arxiv.org/pdf/1805.00754.pdf>.
- Kishimoto, Risako et al. [2017]. “Statistical modeling of global mean wave height considering Principal Component Analysis of sea level pressures and its application to future wave height projection”. In: *Hydrological Research Letters* 11.1, pp. 51–57. DOI: 10.3178/hrl.11.51.
- Kobak, Dmitry et al. [2016]. “Demixed principal component analysis of neural population data”. In: *eLife* 5, e10989. DOI: 10.7554/eLife.10989. URL: <https://elifesciences.org/articles/10989>.
- KrzyÅko, MirosÅaw et al. [2023]. “Spatio-temporal principal component analysis”. In: *Spatial Economic Analysis*. DOI: 10.1080/17421772.2023.2237532. URL: <https://www.tandfonline.com/doi/full/10.1080/17421772.2023.2237532>.
- Laptev, Ivan [2005]. “On space-time interest points”. In: *International Journal of Computer Vision*. Vol. 64. 2–3. Springer, pp. 107–123.
- Liang, Xian, Yuefeng Xu, and Rui Zhang [2022]. *Air-Sea Interaction and Surface Wave Prediction: An Overview*. <https://arxiv.org/pdf/2204.04438.pdf>. arXiv preprint arXiv:2204.04438.
- Margaritella, Nicolo, Vanda Inacio, and Ruth King [2023]. “A Bayesian functional PCA model with multilevel partition priors for group studies in neuroscience”. In: *arXiv*. DOI: 10.48550/arXiv.2312.16739. eprint: 2312.16739. URL: <https://arxiv.org/abs/2312.16739>.
- Massel, Stanislaw R. [2000]. “Wavelet Analysis for Processing of Ocean Surface Wave Records”. In: *Ocean Engineering* 27.3, pp. 231–242. DOI: 10.1016/S0029-8018(00)00044-5.
- Monahan, Adam H. [2000]. “Nonlinear principal component analysis by neural networks: Theory and application to the Lorenz system”. In: *Journal of Climate* 13.4, pp. 821–835. DOI: 10.1175/1520-0442(2000)013<0821:NPCABN>2.0.CO;2.
- Naeini, Saeed Saviz, and Reda Snaiki [2024]. “A novel hybrid machine learning model for rapid assessment of wave and storm surge responses over an extended coastal region”. In: *Coastal Engineering*. DOI: 10.1016/j.coastaleng.2024.104503.
- North, Gerald R et al. [1982]. “Sampling Errors in the Eigenvectors of the Covariance Matrix: Application to Atmospheric Data”. In: *Journal of Climate and Applied Meteorology* 21.12, pp. 214–233.
- Park, Junmo, and Sung-Hyuk Bae [2023]. “Geometric interpretation of principal component analysis and its generalization to correspondence analysis”. In:

- arXiv preprint arXiv:2307.15213.* URL: <https://arxiv.org/abs/2307.15213>.
- Perrin, T. V. E. et al. [2020]. *Functional principal component analysis for global sensitivity analysis of model with spatial output*. arXiv preprint arXiv:2005.10285.
- Pierella, Fabio et al. [2020]. “The DeRisk database: Extreme Design Waves for Offshore Wind Turbines”. In: *arXiv preprint arXiv:2012.02839*. URL: <https://arxiv.org/pdf/2012.02839.pdf>.
- Qin Y Wang Y, Zhang X [2020]. *Spatiotemporal pattern extraction for large-scale ocean forecasting data using low-rank decomposition methods*. URL: <https://doi.org/10.1029/2020JC016321>.
- Ricondo, JosÃ©, Fernando J. MÃ©ndez, et al. [2023]. “HySwash: A hybrid model for nearshore wave processes”. In: *Coastal Engineering*. Combines sampling, clustering, and interpolation with high-fidelity hydrodynamics.
- Salameh, Edward et al. [2019]. “Monitoring Beach Topography and Nearshore Bathymetry Using Spaceborne Remote Sensing: A Review”. In: *Remote Sensing* 11.19, p. 2212. DOI: 10.3390/rs11192212. URL: <https://www.mdpi.com/2072-4292/11/19/2212>.
- Su, Jingyong et al. [2014]. “Statistical Analysis of Trajectories on Riemannian Manifolds: Bird Migration, Hurricane Tracking, and Video Surveillance”. In: *The Annals of Applied Statistics* 8.1, pp. 530–552.
- Tadic, Lidija, Ognjen Bonacci, and Tamara BrlekoviÄ [2019]. “An example of principal component analysis application on climate change assessment”. In: *Theoretical and Applied Climatology* 138.3–4, pp. 1049–1062. DOI: 10.1007/s00704-019-02887-9. URL: <https://doi.org/10.1007/s00704-019-02887-9>.
- Turner, Ian L., Roshanka Ranasinghe, and Andrew D. Short [2014]. “CZM Applications of Argus Coastal Imaging at the Gold Coast, Australia”. In: *Journal of Coastal Research* 30.3, pp. 539–554. DOI: 10.2112/JCOASTRES-D-13-00040.1.
- Viaña-Borja, S. P. et al. [2024]. “Satellite-Derived Bathymetry Using Sentinel-2 in Mesotidal Coasts”. In: *Coastal Engineering* 185, p. 105675. URL: <https://www.sciencedirect.com/science/article/pii/S0378383924001923>.
- Wang, Kun et al. [2014]. “Low-Rank Matrix Estimation-Based Spatiotemporal Image Reconstruction for Dynamic Photoacoustic Computed Tomography”. In: *Proceedings of SPIE - The International Society for Optical Engineering*. Vol. 8943. SPIE, p. 89432I. DOI: 10.1117/12.2041850. URL: <https://doi.org/10.1117/12.2041850>.
- Wilks, Daniel S [2011]. *Statistical Methods in the Atmospheric Sciences*. Academic Press.
- Yan, Jiahe et al. [2021]. “Spatial–Temporal Traffic Flow Data Restoration and Prediction Method Based on the Tensor Decomposition”. In: *Applied Sciences*

- 11.19, p. 9220. DOI: 10.3390/app11199220. URL: <https://doi.org/10.3390/app11199220>.
- Zhang, Jing, Jan Laufer, and Paul Beard [2014]. “Fast spatiotemporal image reconstruction based on low-rank matrix estimation for dynamic photoacoustic computed tomography”. In: *IEEE Transactions on Medical Imaging* 33.6, pp. 1234–1245.