

A Self-Adaptive Method for Mapping Coastal Bathymetry On-The-Fly from Wave Field Video

Gawehn, Matthijs; de Vries, Sierd; Aarninkhof, Stefan

DOI

[10.3390/rs13234742](https://doi.org/10.3390/rs13234742)

Publication date

2021

Document Version

Final published version

Published in

Remote Sensing

Citation (APA)

Gawehn, M., de Vries, S., & Aarninkhof, S. (2021). A Self-Adaptive Method for Mapping Coastal Bathymetry On-The-Fly from Wave Field Video. *Remote Sensing*, 13(23), Article 4742.
<https://doi.org/10.3390/rs13234742>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Article

A Self-Adaptive Method for Mapping Coastal Bathymetry On-The-Fly from Wave Field Video

Matthijs Gawehn ^{1,2,*} , Sierd de Vries ² and Stefan Aarninkhof ²

¹ Unit of Marine and Coastal Systems, Department of Applied Morphodynamics, Deltares, 2600 MH Delft, The Netherlands

² Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN Delft, The Netherlands; Sierd.deVries@tudelft.nl (S.d.V.); S.G.J.Aarninkhof@tudelft.nl (S.A.)

* Correspondence: Matthijs.Gawehn@deltares.nl or m.a.gawehn@tudelft.nl

Abstract: Mapping coastal bathymetry from remote sensing becomes increasingly more attractive for the coastal community. It is facilitated by a rising availability of drone and satellite data, advances in data science, and an open-source mindset. Coastal bathymetry, but also wave directions, celerity and near-surface currents can simultaneously be derived from aerial video of a wave field. However, the required video processing is usually extensive, requires skilled supervision, and is tailored to a fieldsite. This study proposes a video-processing algorithm that resolves these issues. It automatically adapts to the video data and continuously returns mapping updates and thereby aims to make wave-based remote sensing more inclusive to the coastal community. The code architecture for the first time includes the dynamic mode decomposition (DMD) to reduce the data complexity of wavefield video. The DMD is paired with loss-functions to handle spectral noise and a novel spectral storage system and Kalman filter to achieve fast converging measurements. The algorithm is showcased for fieldsites in the USA, the UK, the Netherlands, and Australia. The performance with respect to mapping bathymetry was validated using ground truth data. It was demonstrated that merely 32 s of video footage is needed for a first mapping update with average depth errors of 0.9–2.6 m. These further reduced to 0.5–1.4 m as the videos continued and more mapping updates were returned. Simultaneously, coherent maps for wave direction and celerity were achieved as well as maps of local near-surface currents. The algorithm is capable of mapping the coastal parameters on-the-fly and thereby offers analysis of video feeds, such as from drones or operational camera installations. Hence, the innovative application of analysis techniques like the DMD enables both accurate and unprecedentedly fast coastal reconnaissance. The source code and data of this article are openly available.



Citation: Gawehn, M.; de Vries, S.; Aarninkhof, S. A Self-Adaptive Method for Mapping Coastal Bathymetry On-The-Fly from Wave Field Video. *Remote Sens.* **2021**, *13*, 4742. <https://doi.org/10.3390/rs13234742>

Academic Editor: Dimitris Poursanidis

Received: 19 October 2021

Accepted: 18 November 2021

Published: 23 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: remote sensing; coastal zone; bathymetry; depth inversion; waves; dynamic mode decomposition; on-the-fly

1. Introduction

Observations of near-shore hydrodynamics and bathymetry are used for various purposes: to study and manage the coast [1–3], to update early warning systems [4], to monitor swimmer safety [5–7], for dredging-and-dumping surveillance [8], and military landing operations [9]. An observation of hydrodynamics or bathymetry with areal coverage, a map, is thereby often beneficial if not a prerequisite to recognize relevant spatial details. A straightforward approach to map coastal hydrodynamics and bathymetry is via video-based remote sensing of a wave-field. In comparison to in-situ measurements, video-based remote sensing is less accurate; however, data acquisition is also less labor-intensive, and measurements have high spatial coverage by default.

Different instruments and video-processing methods are used to map hydrodynamics and bathymetry. In terms of instruments, videos may be recorded with stationary cameras [10], aircrafts [11], UAVs/drones [12–16], (navigational) X-Band radars [17–21], or satel-

lites [22,23]. In terms of video-processing methods, various types exist, e.g., [17,20,24–30]. Most widely used among the coastal remote-sensing community are video-processing methods that analyse wave-dispersion properties. They prove applicable to different instruments [31,32] and allow to estimate several coastal parameters. Consecutive frames of a wave-field recording are scanned [20,33] to extract dominant wave frequencies, ω (rad/s), and associated wave lengths and directions via wave-number vectors, \mathbf{k} with $[k_x, k_y]$ (rad/m) [18,21,34–36], but also wave celerity vectors, \mathbf{c} with $[c_x, c_y]$ (m/s) [37–39], near-surface current vectors, \mathbf{U} with $[u, v]$ (m/s) [11,20,40,41], and the apparent depth, d (m) [9,11,17,42–44]. While ω , \mathbf{k} , and \mathbf{c} are retrieved directly from wave patterns [45,46] or their corresponding frequency–wave-number spectra, e.g., [37], (Figure 1 left), \mathbf{U} and d are retrieved indirectly, albeit simultaneously [2,30,47–49], by matching frequency–wave-number spectra with a theoretical wave model (Figure 1 right). This study focused on mapping \mathbf{c} , d , and \mathbf{U} . Vector fields of \mathbf{k} are directly coupled to vector fields of \mathbf{c} via corresponding ω .

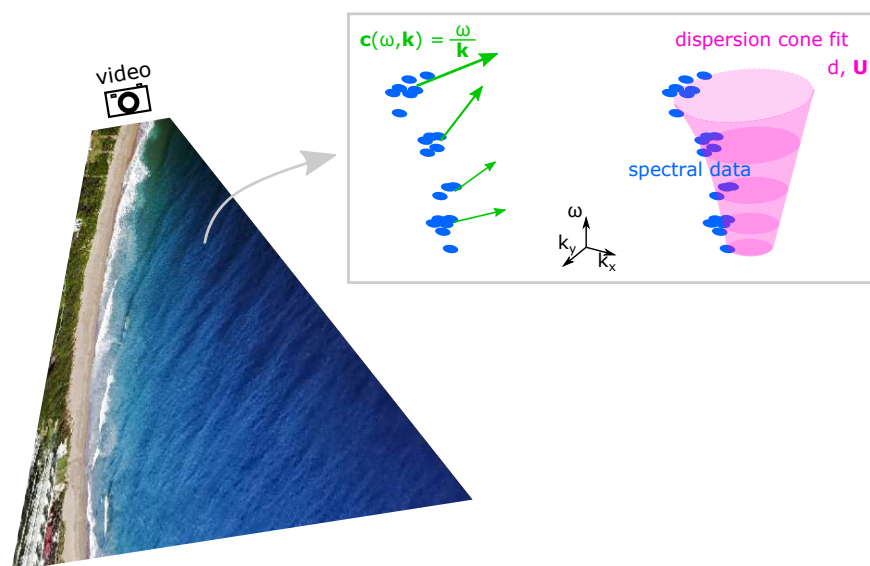


Figure 1. Video of a wave field (left) as a basis to retrieve wave spectra and coastal parameters (\mathbf{c} , d , and \mathbf{U}) (right grey box). Local gravity wave spectra are given by clouds of frequency–wave-number (ω and \mathbf{k}) pairs (blue dots). Local wave celerity vectors \mathbf{c} (m/s) (green) can be calculated from the frequency ω (rad/s) and wave-number vector \mathbf{k} (rad/m). Fitting the Doppler-shifted linear dispersion relationship as a theoretical model (magenta) to the observed spectral data yields local estimates of depth d (m) and near-surface current \mathbf{U} (m/s).

The wave model is typically given by the Doppler-shifted linear dispersion relationship, Equation (1):

$$\omega_{model}(d, \mathbf{U}) = \sqrt{g|\mathbf{k}| \tanh(|\mathbf{k}|d) + \mathbf{U} \cdot \mathbf{k}}, \quad (1)$$

where ω_{model} (rad/s) is the wave model frequency, which is adjusted with unknown parameters d and \mathbf{U} until it optimally matches (and therewith explains) the observed ω . The gravitational acceleration is given by g (m/s²). Note that \mathbf{U} is derived as the depth-averaged velocity of a depth-uniform current profile; however, in natural settings with depth-varying current profiles, \mathbf{U} represents the weighted average of velocities in the upper layer and is therefore typically referred to as the near-surface current vector [20,41]. Advanced techniques also allow to resolve the underlying current profile [50,51]. Yet, the advantage of condensing the profile to a bulk vector \mathbf{U} is a comparably simple expression for the Doppler shift ($+\mathbf{U} \cdot \mathbf{k}$ in Equation (1)), by which \mathbf{U} can be estimated together with d [47,49]. Moreover, near-surface currents can then directly be visualized through maps of vector fields, e.g., [40,52].

Two aspects need to be considered for estimating coastal parameters from video: first, the optimal extraction of wave length and wave frequency characteristics from optical

spectra, and second, the optimal estimation of c , d , \mathbf{U} from the found wave-length and -frequency data. Both aspects are successively treated in Sections 1.1 and 1.2.

1.1. Optimizing Wave-Number–Frequency Extraction from Optical Spectra

One of the core challenges of wave dispersion-based video processing is to robustly identify the wave-number–frequency signature of gravity waves ($\mathbf{k}^\uparrow, \omega^\uparrow$, where \uparrow denotes the “gravity wave signature”). Different strategies [30,37,43,53] follow different pathways to do this (Figure 2). All strategies use greyscale video as a basis and inspect local subdomains to capture spatial variation in \mathbf{k}^\uparrow . Searching for an optimal pathway to retrieve $\mathbf{k}^\uparrow, \omega^\uparrow$, the alternatives are briefly presented (Figure 2). Appendix A contains a more detailed review of how successive transformations in different strategies lead to the retrieval of $\mathbf{k}^\uparrow, \omega^\uparrow$ (Figure 2, arrows) and summarizes some benefits and drawbacks in Table A1.

Local cut-outs from the video form cubes in space time (x, y, t ; in short \mathbf{x}, t) and can directly be used as analysis subdomains. Decomposing a cube reveals the wave constituents of different frequency, either directly in wave-number–frequency space (in \mathbf{k}, ω) [37] (Figure 2, left pathway) or via their complex-valued phase images (in \mathbf{x}, ω) [30] (Figure 2, centre pathway). Separation from the spectral noise floor finally yields $\mathbf{k}^\uparrow, \omega^\uparrow$ (Figure 2 bottom). Associated to one frequency component, a phase image is referred to as a one-component phase image [30] and shows the distinct wave pattern at a certain wave frequency. The suggested benefit of using one-component phase images is better localization of \mathbf{k}^\uparrow . Instead of deriving local one-component phase images (LOCPI) from video cut-outs (Figure 2, centre pathway), global one-component phase images (GOCPI) of the full video domain can be derived [53] (Figure 2, right pathway). These GOCPI are then further analysed in local subdomains. Independent of using LOCPI [30], GOCPI [53], or both [43], the final phase structure should be spatially coherent to get \mathbf{k}^\uparrow .

The benefits of GOCPI are that they can be generated taking global spatial coherence into account [53], while additionally the dimensionality of the video data can be reduced [53]. Reducing dimensionality is an important asset as it offers a reduction in the required computational working memory and thereby an increase in computational speed. For this purpose, a singular value decomposition (svd) (Appendix A, Equation (A1)) can be employed, which is a dimensionality reduction technique that is controlled by the variance in the video. The svd splits the video into modes, consisting of spatial and temporal structures. The first $r \sim O(1) - O(10)$ modes typically describe most of the video’s variance (see σ_j in Appendix A, Equation (A1)) and thereby capture its essence. The remaining modes confine noise and can be discarded, which reduces the data load. Simarro et al. [53] directly use the spatial structures from svd as GOCPI. However, a spatial structure from svd may contain mixed wave patterns of different frequencies (see [54], Figure 6) such that its corresponding temporal structure is not a pure oscillation. Hence, using spatial structures from svd as GOCPI is suboptimal.

Aiming to map coastal parameters on-the-fly, $\mathbf{k}^\uparrow, \omega^\uparrow$ must be retrieved at minimal computational cost. A strategy similar to Simarro et al. [53] is therefore desirable but with pre-knowledge of oscillating spatial structures when generating GOCPI. This is what the dynamic mode decomposition (DMD) aims to do (Appendix A, Table A1).

The DMD is a mathematical procedure designed to reduce the dimensionality of video data by identifying dominant oscillating spatial structures. Different variants of DMD have been developed, e.g., [54–60], employed in various fields of science amongst which electrical engineering [61,62], system and control applications [57], neuroscience [63], but also physical oceanography [58] and coastal engineering [64].

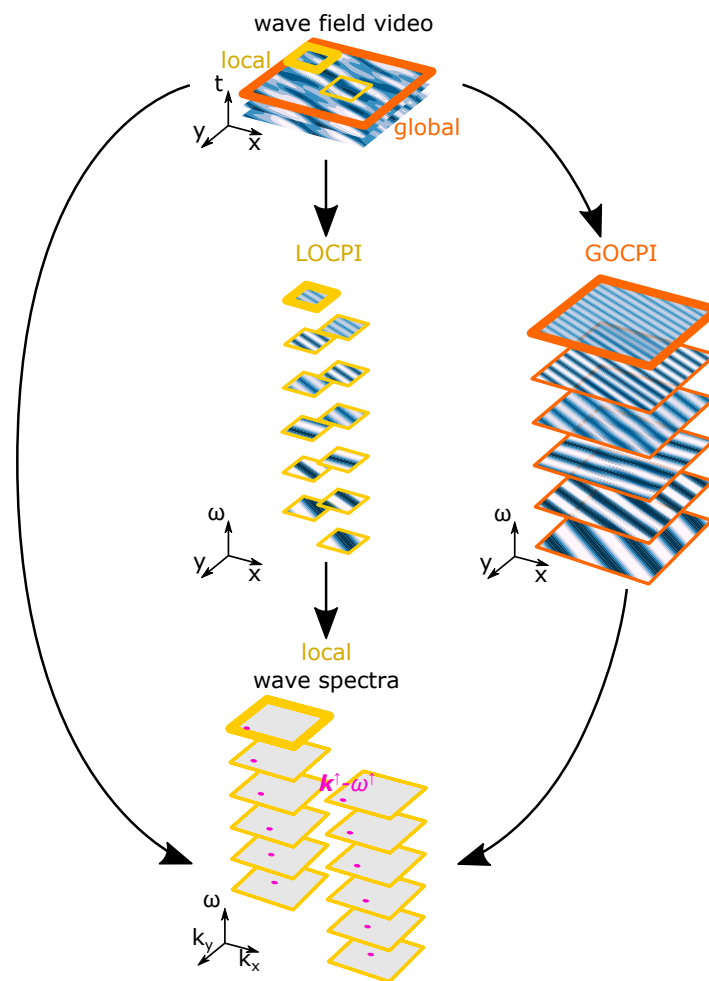


Figure 2. Pathways to identify wave-number–frequency signatures of gravity waves ($\mathbf{k}^\dagger, \omega^\dagger$) (bottom panel, purple dots) in video of a wave field (top panel). The video can be transformed directly from space-time (x, y, t) into the spectral domain (k_x, k_y, ω) (left pathway) or via construction of one-component phase images (x, y, ω) , which may be local (LOCPI) (middle pathway) or global (GOCPI) (right pathway). In the left and middle pathways, all analyses are local (yellow). In the right pathway, the initial construction of GOCPI is global (orange) and then followed by local subdomain analysis to capture spatial variation in $\mathbf{k}^\dagger, \omega^\dagger$.

Here, the spatial structures from DMD represent phase images of waves with different periods. The DMDs' foundation lies in the assumption that snapshots in a video (sequence of frames) are related to each other through a linear dynamic system of oscillatory components. This means that the benefits of Simarro et al. [53], offering GOCPI and the possibility for dimensionality reduction can be complemented with a guarantee that GOCPI oscillate in time. What is more, the oscillation frequencies are inherently found.

To summarize, the DMD can be used to reduce video data to a set of GOCPI (Figure 2, right pathway) but with guaranteed oscillatory time behaviour. These GOCPI are the basis to find local $\mathbf{k}^\dagger, \omega^\dagger$ signatures (Figure 2 left, purple dots): while (global) ω^\dagger are known from the DMD, \mathbf{k}^\dagger still need to be locally deduced. This can be done in various ways, such as via characteristic spatial phase differences in sub-domains [30,43,53] or alternatively via (computationally cheap) spatial 2D-FFTs or particle image velocimetry (PIV) [23,65]. Derived local $\mathbf{k}^\dagger, \omega^\dagger$ form the basis to retrieve maps of wave celerity per GOCPI frequency, via $c(\omega^\dagger) = \omega^\dagger / \mathbf{k}^\dagger$ [66], but also maps of depths and near surface currents d, \mathbf{U} through Equation (1).

1.2. Optimizing Depth and Near-Surface Current Estimates

Pursuing the best approximations to d and \mathbf{U} , optimization problems using Equation (1) (with/without Doppler shift $+\mathbf{U} \cdot \mathbf{k}$) have been stated differently, with some as (non-linear) least-squares (LS) minimization problems [20,30,43] and others as maximization problems of a normalized scalar product (NSP) [19,49]. These schemes have been formulated to handle an abundance of spectral information from Fourier decompositions. However, applying dimensionality reduction techniques to video data concentrates spectral information to its essence offering far fewer data points for an optimization of d , \mathbf{U} . As such, inaccuracies and outliers in the spectral data gain negative influence on the solution. Standard LS minimization is not fit to handle this issue and with little data to fit, NSP maximization using a Heaviside step function is a crude approach. Outlier contamination is a common issue in applications such as the training of neural networks [67], because reducing the relatively large residual of an outlier is more attractive in minimizing the cost function than reducing the small residual of an inlier. In standard LS optimization, residuals are squared, whereby the impact of outliers on the solution is disproportionately large. However, LS problem statements have the benefit that many different methods have been developed to solve them [68], among which the Levenberg–Marquardt method is often applied to invert d , \mathbf{U} e.g., [2,43]. It is therefore attractive to adapt LS formulations such that they can handle outliers. This is achieved using various kinds of loss functions [69], among which the Cauchy (also called Lorentzian) loss function is an effective type.

Higher accuracy in d , \mathbf{U} can also be achieved by successively producing estimates and converging the results with a Kalman filter [29,43,70]. The quality of an estimate is thereby typically judged based on the sensitivity of the fit to the data. If the data are plagued by inaccuracies and outliers, this sensitivity increases, making the Kalman filter a suitable tool to mitigate their influence.

To summarize, while loss-functions can increase the quality of an individual d , \mathbf{U} estimate, the application of a Kalman filter increases the quality over several d , \mathbf{U} estimates. Both techniques can be used simultaneously.

1.3. Outlook of This Study

This study is about a self-adaptive, robust method to map c , d and \mathbf{U} on-the-fly from video of a wave-field. The DMD plays a key role in making the video-processing algorithm self-adaptive to the data and computationally fast: it reduces (video) data complexity, finds the dominant wave-components, and allows self-adaptive sampling schemes, which cause the standardly used computational cubes to instead become pyramids for optimal localization. For the optimization of d , \mathbf{U} , the algorithm implements a loss-function to handle spectral outliers, which seemingly counteracts the error of overestimating \mathbf{U} and also reduces depth errors. The two errors are commonly interlinked [2,47,48,70]. The algorithm temporarily stores spectral data and employs Kalman filtering for quick convergence of measurements, and it comprises a set of rules and filters for autonomous decision making such that the algorithm does not need to be tuned to the field site. In summary, the algorithm's main elements include:

- Data reduction and retrieval of wave components through DMD.
- Wave-component-dependent subdomains using pyramid cells.
- Counteracting spectral outliers and errors in d, \mathbf{U} with loss functions.
- Fast convergence of d, \mathbf{U} and recognition of current refraction through temporary spectral data storage.
- Additional fast convergence of d, \mathbf{U} using Kalman filtering.

Section 2 describes the workflow of the proposed video-processing algorithm, which includes explanations of the core principles of the DMD and the workflow of the algorithm. Section 3 describes the field sites and data. In Section 4, the algorithm is validated for measuring near-shore bathymetry (i.e., d) based on four field sites in the USA, the UK, the Netherlands, and Australia. A qualitative discussion on the algorithm's ability to

measure hydrodynamics (i.e., \mathbf{c} , \mathbf{U}) follows in Section 5, together with a discussion on the algorithm's potential for mapping on-the-fly. The findings are concluded in Section 6.

2. Method

Ahead of presenting the mapping algorithm in Section 2.2, first the implemented dynamic mode decomposition (DMD) is explained and demonstrated in Section 2.1.

2.1. Dynamic Mode Decomposition

The DMD forces oscillatory time dynamics through a set of discrete linear differential equations whose solution consists of complex eigenvalues and eigenvectors. The eigenvalues represent the temporal oscillations, which may include a real part denoting growth or decay. The corresponding eigenvectors are the dynamic modes and represent spatial structures, which after entry-wise normalization to unity represent global one-component phase images (GOCPI).

Suppose a linear model \mathbf{A} can advance a (squeezed) frame \mathbf{x}_n at time t_n to the next frame at time t_{n+1} : $\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n$. Extending the model to advance a (time) series of N frames simultaneously, $\mathbf{Y} = \mathbf{A}\mathbf{X}$, where matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N-1}]$ and $\mathbf{Y} = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N]$ pair each frame \mathbf{x}_{n+1} with the previous frame \mathbf{x}_n . It essentially requires \mathbf{A} to propagate a frame through time since $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots] = [\mathbf{x}_1, \mathbf{A}\mathbf{x}_1, \mathbf{A}^2\mathbf{x}_1, \dots]$, which in mathematical terms is referred to as a Krylov sequence. It indicates that $\mathbf{Y} = \mathbf{A}\mathbf{X}$ is a discrete formulation that is closely tied to a system of continuous differential equations $d\mathbf{x}/dt = \mathbf{A}\mathbf{x}(t)$ and therewith an eigenvalue problem $\lambda\boldsymbol{\varphi}e^{\lambda t} = \mathbf{A}\boldsymbol{\varphi}e^{\lambda t}$. Conceptually, \mathbf{A} hence describes a dynamical process, whose eigenvectors $\boldsymbol{\varphi}$ present the dynamic modes, with the associated frequencies captured in the complex eigenvalues λ . Note that, in contrast to the modes retrieved from an svd (Section 1, [53]), dynamic modes do not have to be orthogonal to each other, which implies that they do not have to be fully independent of each other.

The goal of the DMD is to find the eigenvalues and eigenvectors of \mathbf{A} at minimal computational cost. Finding the eigenvalues and eigenvectors straightforwardly by first calculating $\mathbf{A} \approx \mathbf{Y}\mathbf{X}^\dagger$ ($\dagger \equiv$ pseudo-inverse) poses a problem for computer memory. Say video footage has a frame size of 1000×1000 px, then \mathbf{X} and \mathbf{Y} have a row size $m = 10^6$, resulting in an \mathbf{A} matrix of size $m \times m = 10^6 \times 10^6$. Even in case a computer can handle such data loads, the calculation is slow, and the matrix size suggests a large amount of redundancy. The conceptual idea is to convert the eigenvalue problem from m into a lower dimension r . Typically, r is in the range 1–100 such that $r \ll m$, expressing a severe dimensionality reduction. By deflating the eigenvalue problem $m \rightarrow r$, eigenvectors and eigenvalues can be quickly calculated. Subsequently, the eigenvectors are inflated again $r \rightarrow m$ to yield the dynamic modes. Although the details of the conversions differ between DMD algorithms (e.g., standard DMD vs. optimized DMD [60,71], respectively), they share the common principle of using r modes from the svd of \mathbf{X} (Appendix A, Equation (A1)) for dimensionality reduction. There exists no general strategy to make a choice for r . It may be based on existing knowledge about the observed system [60], but it can also be based on the amount of singular values needed to capture a certain percentage of the variance in the video, for example, 99% [54], or on an algorithmic truncation of noise [72].

Note that \mathbf{X} (and \mathbf{Y}) may first be converted to a time-analytic signal [53]. It is not necessary, but it has the benefit that the DMD, similar to an FFT, does not produce conjugate dynamic mode pairs. Then, r frequency components associate to r modes instead of $2r$ modes. Preventing the generation of conjugate modes also prevents the DMD to produce them imperfectly (i.e., with slightly different frequency to their counterpart). Analytic extension comes at a computational cost; however, since r can be halved to achieve the same number of frequency components, successive matrix multiplications within the DMD become computationally cheaper.

An important factor when applying a DMD is whether the recorded data contain standing-wave behaviour. Without adjustments, a DMD cannot capture standing waves [55]. For a fixed framerate, Tu et al. [55] offer a straightforward solution through augmentation

of the video matrix \mathbf{X} with a time-shifted version of itself, by which the DMD acquires the skill to detect standing waves.

From the existing DMD algorithms, the *optimized DMD based on variable projections* [60] is an elaborate variant. Formulated as a least-squares optimization problem, it is more accurate than other DMD algorithms and theoretically allows video frames to be spaced non-equidistantly in time. Instead of splitting the video into two video matrices \mathbf{X} and \mathbf{Y} , it uses a single, transposed video matrix $\mathbf{X}^T = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$. A detailed explanation can be found in Appendix B. The methods skill for complex harmonic analysis was recently demonstrated in modelling rotating detonation waves [73]. Note that other DMD algorithms could be potent alternatives, for example, by allowing elaborate forms of regularization to handle noisy or lower-quality images [74].

Wave components in a wave field can be accurately extracted using the DMD [54]. To illustrate this, a wave field consisting of six known wave components is considered and recorded over a period of 32 s at 2 fps. The corresponding video matrix hence comprises 64 frames (\mathbf{X}^T has 64 rows, see Appendix B, Equation (A2)). Subsequently, the signal is made time-analytic such that r modes associate to r frequency components. The DMD with $r = 6$ modes identifies the underlying components precisely (Figure 3a, orange stars) (Figure 3b). It demonstrates that the DMD is not only powerful in analysing a short frame sequence but therein superior compared to a standard time-domain fast Fourier transform (FFT) (Figure 3a, green squares): the FFT returns (much) more spectral data and with significant redundancy, and yet, the pre-set frequency resolution restricts its ability to capture the six intrinsic wave components to a mere rough spectral representation. Half of the components (Figure 3a, first, third, and fifth component) are poorly captured in frequency and amplitude.

For a real wave field, the amount of wave components is not known a-priori and a choice needs to be made for the amount of modes r returned by the DMD. The question arises what the DMD returns if the choice for r is smaller or larger than the actual number of components in the wave field: if r is smaller, the DMD simply returns fewer components, but those components are still correctly represented (Figure 3c, $r = 3$). If r is larger, the DMD still identifies the intrinsic components; however, it also adds spurious modes and these modes may be energetic, which indicates that r being too large is a scenario that should be avoided (Figure 3c, $r = 12$). For observations of a real wave-field this scenario is unlikely, as waves have a stochastic nature and typically dense spectra [66]. In fact, for $r \sim O(1) - O(10)$, the number of modes is certainly less than the actual number of wave components (cf. Figure 3c, $r = 3$), and the DMD's representation of the wave field simplifies to the governing wave components.

A real wave field may also experience standing-wave dynamics as waves reflect back from the shoreline. Although, at flat dissipative beaches, sea-swell waves are typically progressive [75], at steeper beaches and longer sea-swell periods, incident wave reflection can be significant with reflection coefficients up to $K = 0.4 - 0.45$ [76–79]. In the presence of hard structures like sea-walls even up to $K \approx 0.9$ [80]. Here, $K = [0, 1]$ denotes [no, full] reflection [81]. High wave reflection signals (partially) standing-wave characteristics, prompting adaption of the DMD as of [55]. Doing so enables the DMD to cope with any degree of incident wave reflection (Figure 3c, $K = 0$, $K = 0.5$, and $K = 1$) and thereby accommodates application to a broad range of wave field scenarios.

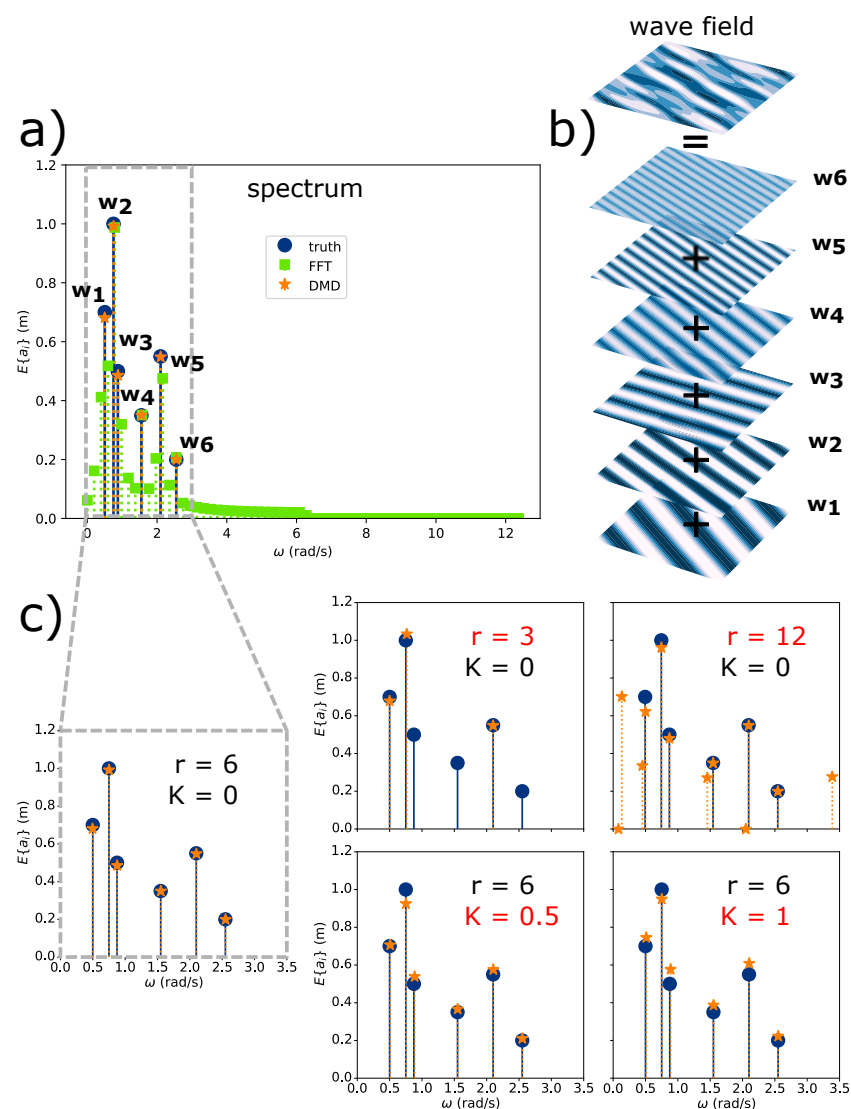


Figure 3. Dynamic Mode Decomposition (DMD) of an artificial wave field with six wave components ($w_1 - w_6$), recorded over a period of 32 s at 2 fps, resulting in 64 video frames. (a) True amplitude spectrum of the wave field (blue dots), compared against spectra acquired from DMD with $r = 6$ modes (orange stars) and FFT (green squares); $E\{a_i\}$ (m) denotes the expected amplitude of the i th wave component and ω (rad/s) the angular frequency. (b) Real part of the complex valued dynamic modes acquired from DMD, resembling the true wave components. (c) The DMD from (a), with $r = 6$ and a progressive wave field with no wave reflection (reflection coefficient $K = 0$) is used as reference (dashed grey outline) in a comparison with DMDs of the same progressive wave field but where the number of modes is halved ($r = 3$, red) or doubled ($r = 12$, red). If $r > 6$, spurious modes appear. The reference case is also compared against DMDs with the same number of modes ($r = 6$) but of wave fields with mixed progressive standing waves ($K = 0.5$, red) or fully standing waves ($K = 1$, red). To acquire the same expected amplitudes as in the reference case, wave components for the two cases where $K > 0$ were rescaled in amplitude, accounting for their nodal structures in space (see also [81], Equation (11)).

2.2. Mapping Algorithm

The proposed workflow for a self-adaptive and on-the-fly mapping algorithm of coastal hydrodynamics and bathymetry follows a series of steps (Figure 4, labels ①...⑬). The workflow requires video in greyscale and orthorectified format as basis input. If the video is in colour, standard BGR-to-grey conversions can be used to prepare the video. The orthorectification differs per field site and is briefly treated in Section 3. The workflow is

set-up in such a way that a video feed could be processed. For that purpose, the algorithm marches forward in time by consecutively analysing short sequences of N video frames (e.g., $N = 64$) and then updating the maps of c , d and \mathbf{U} after each sequence, which is referred to as mapping updates. Consecutive frame sequences may overlap, as for on-the-fly application, which is explained in Section 5 (see also [82], Figure 3). Default settings for the algorithm as used in this study are listed in Table A2 (Appendix C).

The workflow commences with the global analysis of a video frame sequence. The first step is to retrieve r global one-component phase images (GOCPI) through DMD (Figure 4, ①). GOCPI linked to frequencies outside the ocean wave band (Appendix C, Table A2, $[T_{min}, T_{max}]$) are discarded (Figure 4, ②). (Note that in case the video is not extended to a time-analytic signal, conjugate modes are also discarded; see Section 2.1). By construction of the DMD, the remaining number p of GOCPI with frequencies $\omega_1^\uparrow \dots \omega_p^\uparrow$ describe dominant wave field components. These are treated as equally important (regardless of their spectral weight b ; see Appendix B).

Knowing ω^\uparrow , the maximum wavelengths in each GOCPI are predictable. The size of the subdomains that are used to determine local \mathbf{k}^\uparrow , can thereby be automatically tailored to the individual GOCPI (Figure 4, ③). This is done ahead of analysing any location. A conservative rule is to set the subdomain size to one or two offshore wave lengths, $L_{off}(\omega^\uparrow) = 2g\pi/(\omega^\uparrow)^2$, where g denotes the gravitational acceleration. Here, $2 \times L_{off}(\omega^\uparrow)$ is used, unless this is too large, for example, near a boundary, where the size is reduced up to a minimum of $1 \times L_{off}(\omega^\uparrow)$. High-frequency GOCPI are analysed with smaller subdomains than low-frequency GOCPI, such that stacking the subdomains in layers on top of each other creates a pyramid-shaped cell at a certain location (see Figure 4, ③, yellow pyramid). Note that this contrasts with the usual cube shape, whose size is typically predefined manually, e.g., [41,83].

Since the pixel resolution is constant between cell layers, large layers used for lower frequencies likely encompass plentiful pixels, capturing the underlying waves in unnecessarily high spatial resolution. It is therefore computationally attractive to subsample larger cell layers. The rules for subsampling require a minimum resolution of 8 samples per L_{off} . Instead of skipping pixels for subsampling, the current set-up lowers the resolution through fast bilinear interpolation. It preserves more image information and makes the algorithm robust to videos with different pixel resolutions. For proper analysis, a cell layer is set to hold at least 24×24 samples. Note that demanding eight samples per offshore wave length may exclude higher-frequency GOCPI from being analysed, if the pixel resolution of the video is not high enough to capture the correspondingly short wavelengths.

After subdomain sizes and resolutions have been determined, the *local* analysis of the GOCPI commences. The local analysis occurs around points in a grid. The processing of different grid point locations can be done in parallel, which increases computational speed. Since the algorithm aims to map results on-the-fly, the grid resolution is determined based on computational speed (see also Section 5). Note that computation speeds are different for different processing machines. A solution to find an optimal number of gridpoints for different processing machines could be to start with a low number of gridpoints and then increase the number of gridpoints for consecutive frame sequences until an optimum is reached.

To analyse a certain grid point, first a pyramid-shaped cell is built around it (Figure 4, ④) using GOCPI subdomains as cell layers with formats as determined under step ③. Each cell layer is first autocorrelated in space to accentuate the waveform and tapered with a two-dimensional Hanning-window, which focusses wave information to the centre point and prepares for analysis with two-dimensional fast Fourier transforms (2D-FFTs). For robustness, \mathbf{k}^\uparrow (Figure 4, ⑤) is estimated in two different ways: directly, by analysing spatial properties through 2D-FFT and indirectly, by analysing wave celerity through particle image velocimetry (PIV).

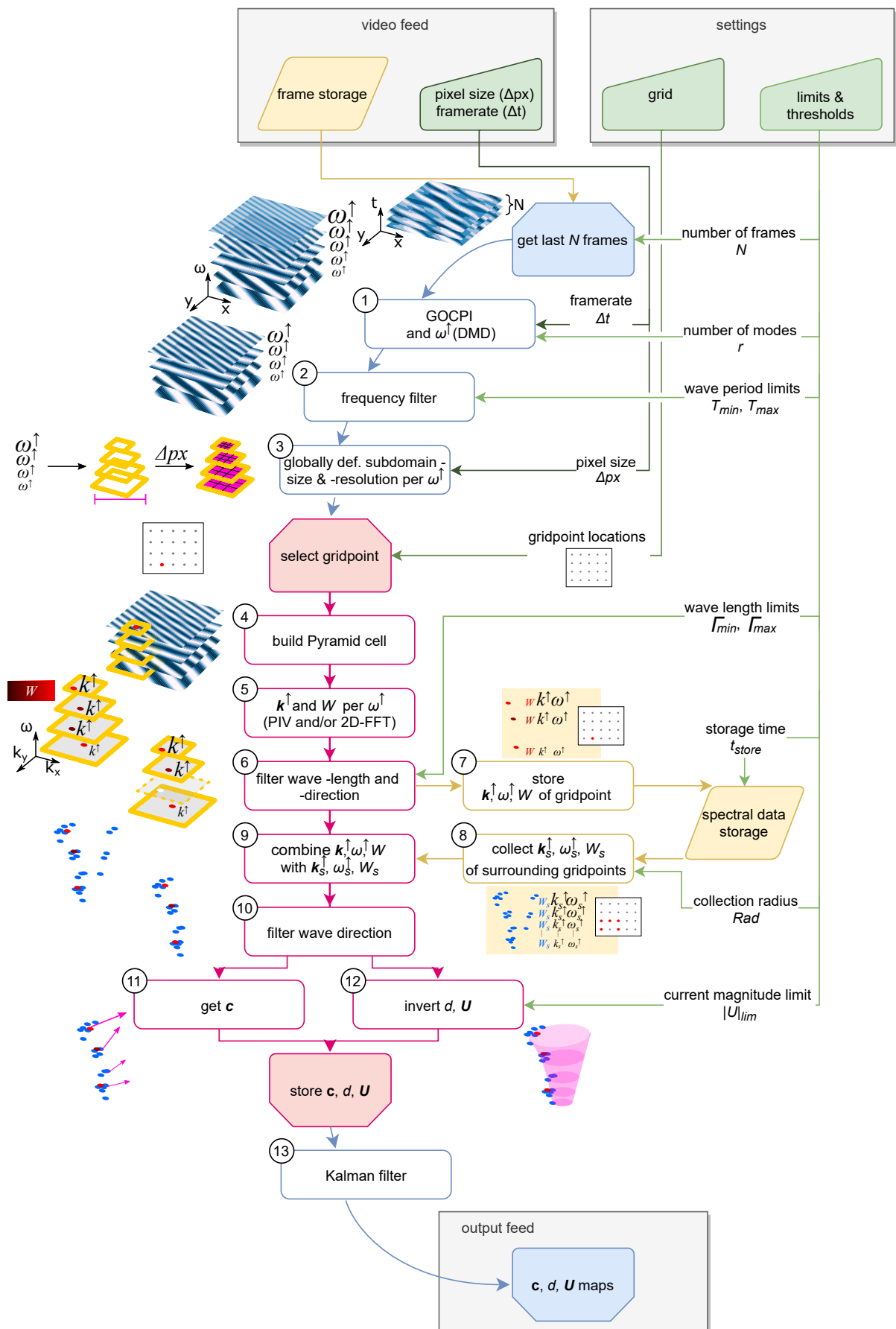


Figure 4. Workflow for mapping coastal hydrodynamics and bathymetry on-the-fly from video of a wave field. Steps in the flowchart are visualized with icons. Box shapes denote: data (parallelogram), input (right trapezoid), process loop start (trimmed top corners) and process loop end (trimmed bottom corners), and process (rectangle). Box and arrow colours relate to: storage (gold), input (green), for-loop (blue), and parallel-for-loop (magenta). Arrows and their annotations signify flow of information. The input requires video with a top-down view, its pixel resolution, and framerate. Computational grid and other settings suffice with default values. The output contains maps of wave directions and celerity, depth, and near-surface currents (grey square bottom row). Symbols represent: $GOCP1$ = global one-component phase images, ω^\uparrow (rad/s) = wave frequency per $GOCP1$, \mathbf{k}^\uparrow (rad/m) = wave-number vector at gridpoint, W (-) = weight per $\mathbf{k}^\uparrow, \omega^\uparrow$ pair, \mathbf{c} (m/s) = wave celerity vector at gridpoint, d (m) = depth at gridpoint, \mathbf{U} (m/s) = near-surface current vector at gridpoint, subscript s = from spectral data storage. Other symbols are labelled with arrow annotations.

Performing a 2D-FFT on a cell layer yields its spectral content in space. Besides the footprint of a gravity wave component, \mathbf{k}^\uparrow , this also includes spectral noise. Typically, an energy threshold aims to separate the two [41]. To avoid a search for an optimal energy threshold [2], here simply the spectral point with maximum energy is chosen as \mathbf{k}^\uparrow . A different approach to acquire \mathbf{k}^\uparrow is via the wave celerity, $\mathbf{k}^\uparrow = \omega^\uparrow / \mathbf{c}^\uparrow$. The wave celerity is encoded in the complex values of the cell layer. The difference between the real and imaginary parts denotes a temporal phase shift of 90° , hence resembling a quarter wave period. Performing PIV between the real and imaginary image of the cell layer yields the translation of (wave) patterns over a quarter wave period, which directly translates to \mathbf{c}^\uparrow . Note that the temporal phase shift between real and imaginary parts is 90° regardless of whether the video matrix \mathbf{X} is analytic or whether dealing with standing waves. While the 2D-FFT approach inherently presumes that the observed pattern in a cell layer describes a wave, the PIV approach does not, as it investigates movement of any pattern in the cell layer. Both approaches may have their benefits in case the pattern is noisy and are therefore used synchronically. If one or both estimates are unphysical, they are discarded in following the filter steps. Each estimate of \mathbf{k}^\uparrow gets a weight W assigned, whose value between [0, 1] gives an indication of the respectively [low, high] quality of the estimate (Figure 4, ⑤ red-coloured bar). A weight, W , is calculated from the correspondence between two images: the source image and the target image. For the 2D-FFT approach, this is the correspondence between the original pattern displayed by the cell-layer and the approximation of this pattern by its most energetic spectral wave component. For PIV, it is the correspondence between the translated real image of the cell-layer with the imaginary image of the cell-layer. The correspondence is calculated via $W = 1 - \varepsilon_{im}$, where ε_{im} represents the normalized root-mean-square error between the source image relative to the target image [84].

Together, the estimates of \mathbf{k}^\uparrow over all ω^\uparrow layers in the pyramid cell form a sparse spectral point cloud (SSPC) of $\mathbf{k}^\uparrow, \omega^\uparrow$ pairs with assigned weights W . The SSPC expects to follow a cone shape as described by the dispersion relationship (Figure 1; Equation (1)). This pre-knowledge aids in the identification and removal of unphysical $\mathbf{k}^\uparrow, \omega^\uparrow$ points in the SSPC by means of a wavelength and direction filter (Figure 4, ⑥). Wave lengths are filtered using their ratio over the offshore wave length $\Gamma = |\mathbf{k}_{off}| / |\mathbf{k}^\uparrow|$, where $|\mathbf{k}_{off}| = 2\pi / L_{off}(\omega^\uparrow)$. SSPC points with $\Gamma < 0.3$ occupy the shallow water regime as they resemble wavelengths larger than twenty times the local depth [85] and are thereby too large to capture morphological detail. Such \mathbf{k}^\uparrow are deemed unsuited for a localized depth estimate, d . On the other hand, SSPC points with $\Gamma > 1.0$ signalize an unphysical deep water regime if \mathbf{U} is small and are therefore also deemed unsuited. Note that in other algorithms this upper limit is often set lower, to $\Gamma > 0.9$, which presents an approximate elbow value where the uncertainty in estimates of d becomes disproportionately large [39,53,86]; however, a higher limit $\Gamma > 1.0$ preserves spectral points that are valuable for the estimation of \mathbf{U} and thereby also simultaneous estimates of d . Lastly, the wave direction filter excludes SSPC points, which do not align with the general direction by means of the svd filter of Gawehn et al. [2].

The idea behind the next two steps of the workflow (Figure 4, (7), (8)) is to augment the SSPC left after (6) with additional spectral points to make it a dense spectral point cloud (DSPC), which captures directional spread in time and space and thereby allows for a solid inversion of d, \mathbf{U} . As desired and designed, the SSPC as is holds the essence of the local hydrodynamics. The sparsity, however, brings along statistical uncertainty in time and space. An approach to acquire a DSPC is to combine several SSPCs into one. The additional SSPCs can be retrieved from preceding updates (i.e., previously analysed frame sequences) but also from surrounding grid locations within a collection radius Rad (Figure 4, (8) attached green arrow; Appendix C, Table A2). Such an approach requires memorizing SSPCs in a designated short-term storage. Therein, the just-retrieved SSPC is stored (Figure 4, (7)) and SSPCs from preceding updates and from surrounding grid locations are called up (Figure 4, (8)). The size of the short-term storage for SSPCs is manageable since, by construction, each SSPC holds just a few essential spectral points. Note that accumulating spectral information over successive updates is only a valid approach if their computations occur (near) on-the-fly, because it assumes that the wave signal is stationary across several updates. Hence, SSPC data are discarded after a short stationary time period (e.g., 60 s; see Appendix C, Table A2) and replaced by new SSPCs.

An example illustrates the augmentation of an SSPC to a DSPC by means of stored spectral data: assuming a wave signal is stationary for 1 min, the algorithm stores SSPCs for 1 min. If the processing of a frame sequence takes 15 s, the storage contains *stored* SSPCs of all grid locations from the preceding $60\text{ s}/15\text{ s} = 4$ updates. Now, as the algorithm starts to analyse a certain location, it retrieves one *new* SSPC for that location. By itself, this new SSPC might be sufficient to estimate d but not \mathbf{U} . For that, a DSPC is required, which is achieved by augmenting the new SSPC with stored SSPCs from the same grid location and surrounding grid locations (Figure 4, (9)), be it 10 within some radius Rad . The acquired DSPC then consists of $4_{stored} \times (1_{gridloc} + 10_{surr.gridlocs}) + 1_{new} \times 1_{gridloc} = 45$ SSPCs. In this example, the augmentation of the new SSPC of step (6) has hence produced a DSPC that is 45 times denser.

Using surrounding grid locations to augment spectral information has benefits and drawbacks. A drawback is that depth estimates d become less localized. A benefit is that wave refraction caused by near surface currents \mathbf{U} is captured in space, which is essential for estimating \mathbf{U} (and thereby also improves d). To minimize the loss in localization of d , but to still have improved \mathbf{U} estimates, the SSPCs that make up the DSPC are weighted differently. These SSPC weights are multiplied with the individual quality weights W of the spectral points and are subsequently normalized to the range $[0, 1]$. In the current set-up, the new SSPC at a certain grid location weighs 50%. Stored SSPCs of the same grid location weigh together 25%, and stored SSPCs from surrounding grid locations weigh together 25%. In total, this means that 75% of the spectral information focuses on the currently analysed grid location, and 25% focuses on the surrounding area, thereby aiming to keep d estimates localized but still constructing a DSPC that includes enough current refraction for \mathbf{U} estimates.

The DSPC from step (9) may include a minority of spectral points from incorrect wave directions. This occurs because some of the SSPCs, which the DSPC consists of, contain too little spectral information to determine wave directions at step (6). Therefore, the direction filter repeats for the DSPC (Figure 4, (10)). Now, the DSPC is ready for the algorithm to retrieve wave celerities per frequency, $c^\uparrow(\omega^\uparrow)$ (Figure 4, (11)) and depths and surface currents d, \mathbf{U} (Figure 4, (12)). While c^\uparrow are directly computed via $c^\uparrow = \omega^\uparrow/k^\uparrow$, the inversion of d, \mathbf{U} is done by fitting a wave model—here, the Doppler-shifted dispersion relationship (Equation (1))—to the spectral points of the DSPC. The fit results from a nonlinear regression that aims to minimize the sum of residuals between the model frequencies $\omega_{j,model}$ (Equation (1)) and the observed frequencies ω_j^\uparrow per spectral point j using Equation (2). The implicit link of observed k_j^\uparrow with $\omega_{j,model}$ and ω_j^\uparrow is assumed to be trivial.

$$\begin{aligned}
\text{minimize } F(d, \mathbf{U}) &= \frac{1}{2} \sum_j W_j \rho(f_j(d, \mathbf{U})^2) \\
&\text{subject to} \\
&d_{min} < d < d_{max} \\
&|\mathbf{U}| < |\mathbf{U}|_{max} \\
&\text{with} \\
\rho(f_j(d, \mathbf{U})^2) &= \alpha^2 \ln \left(1 + \frac{f_j(d, \mathbf{U})^2}{\alpha^2} \right) \\
&\text{and} \\
f_j(d, \mathbf{U}) &= \omega_{j,model}(d, \mathbf{U}) - \omega_j^\dagger,
\end{aligned} \tag{2}$$

where $F(d, \mathbf{U})$ is the cost function, minimized by adjusting the regression parameters d and \mathbf{U} . Typically, residuals per spectral point, j , are evaluated by the product of their weight W_j with the difference $f_j(\cdot)$ between $\omega_{j,model}$ and ω_j^\dagger . Here, $f_j(\cdot)$ is first modulated by a Cauchy loss function $\rho(\cdot)$ to penalize outliers. A predefined softmargin α tunes $\rho(\cdot)$ to the optimization problem. Setting α such that the Cauchy loss function heavily penalizes residuals larger than $\sim 1/5$ th of the average DMD frequency resolution produces accurate d , while counteracting overfitting of \mathbf{U} . Lastly, bounds for depth $[d_{min}, d_{max}]$ and surface current magnitude $|\mathbf{U}|_{max}$ set the range in which a solution for d, \mathbf{U} is sought. To improve estimates, first a regression is done without a Doppler shift to acquire a close estimate for d that can be used as an initializer for the follow-up regression including a Doppler shift. Adding the first regression has little impact on computation times, since its estimate of d closely approximates the local minimum of the second regression. The minimization of Equation (2) occurs using a sequential least-squares quadratic programming method (SLSQP) [87], which omits potentially expensive computations of Hessians and allows for straightforward implementation of loss-functions.

The final step of the mapping algorithm is a Kalman filter (Figure 4, (13)) that judges the quality of d, \mathbf{U} results based on the sensitivity of the fit with regard to earlier updates. It causes d, \mathbf{U} estimates to quickly converge over successive updates. The implementation of the Kalman filter is identical to Gawehn et al. [2]. If the observation periods are short, say, less than 5 min, morphodynamics but also hydrodynamics can be assumed stationary, such that process variance is negligible. If observation periods are longer, say, 10–20 min, it may become important to capture changes in surface current direction (e.g., due to the formation of a rip current). To allow for such applications, the Kalman filter assumes small process variances Q_c and Q_U for phase celerities and near-surface currents, respectively (i.e., $Q_c = 0.0005 \text{ m}^2/\text{s}^3$, $Q_U = 0.0005 \text{ m}^2/\text{s}^3$; see Appendix C, Table A2).

3. Field Sites and Data

Videos of four different field sites around the world were used to test the algorithm's performance (Figure 5): Duck (North Carolina, USA) [43], Porthtowan (UK) [88], Scheveningen (NL) [89], and Narrabeen (AU) [90]. Specifics on video collections are listed in Table 1. Cameras were positioned at heights of 43–110 m to observe waves over a large distance without wave-shadowing effects. The recorded videos are available in orthorectified format. The geometries have been solved using ground control points (GCPs), by matching GCP image coordinates with world coordinates. The accuracy of the geometries differs per site and was quantified by the GCP reprojection error in world coordinates. These reprojection errors are generally in the (sub)meter range, but in case of Scheveningen and Narrabeen, they increased to on average ~ 7 m at distances of 0.5–2 km from the camera. At Porthtowan, the reprojection errors are unknown, yet slight errors in the geometry likely exist, particularly further afield [91]. The videos from Duck and Porthtowan (UK) are

~ 17 min long and were recorded with Argus stations. The videos of Scheveningen and Narrabeen are ~ 9 min long and were recorded with UAVs. All videos have a framerate of $\Delta t = 2$ fps. Pixel resolutions are $\Delta px = 5$ m for Duck and Porthtowan, and $\Delta px = 2$ m at Scheveningen and Narrabeen. The hydrodynamic conditions varied over the sites between $H_s = 0.8 - 1.63$ m and $T_p = 5.0 - 10.0$ s (Table 2).

The videos of all sites were analysed using sequences of 64 frames equalling 32 s. Marching forward in time, the next frame sequence overlapped 50% with the previous sequence (i.e., overlap of 32 frames at 2 fps), resulting in mapping updates every 16 s of video. This overlap was fixed for reproducibility, since computation times differ per processing machine. For on-the-fly applications, the overlap varies based on processing speed. This is simulated in Section 5 using a standard laptop. Each image sequence was decomposed into $r = 16$ dynamic modes. The chosen r represents a balance between significantly reducing dimensionality, while retrieving enough pyramid cell layers to estimate d, \mathbf{U} . Details on other settings are found in Appendix C, Table A2.

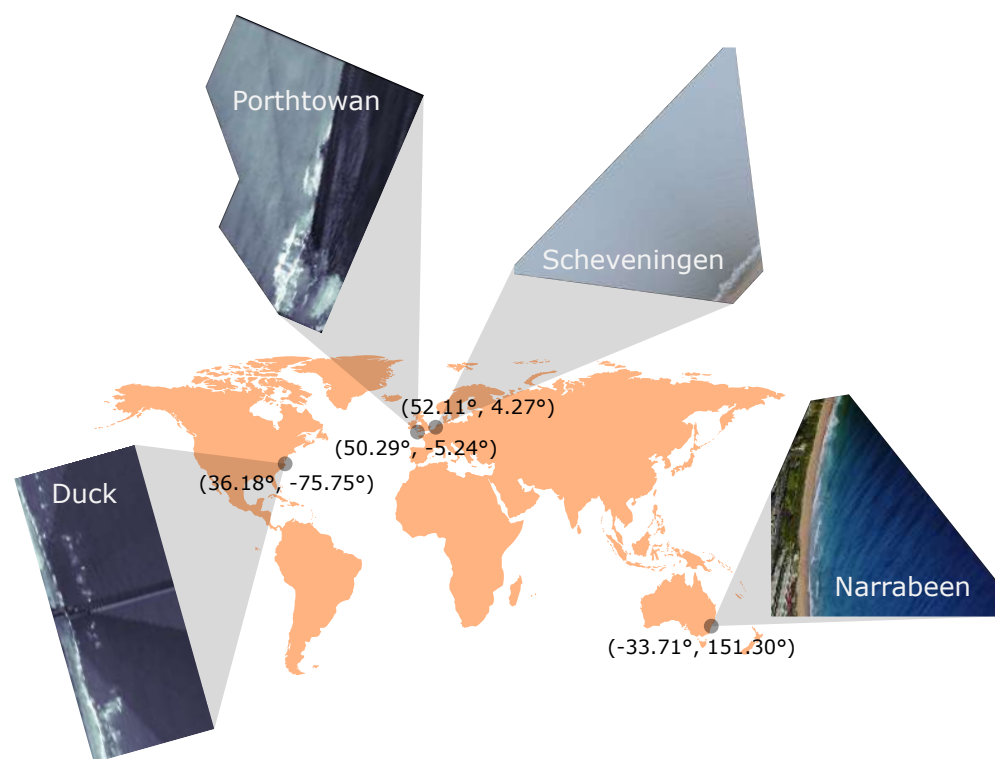


Figure 5. World locations (Lat°, Lon°) of video recordings from Duck (North Carolina, USA), Porthtowan (UK), Scheveningen (NL), and Narrabeen (AU). Videos were recorded with different instruments with different camera properties and therefore have different lighting, format, and orientation.

Table 1. Video collection.

Field Site	Instrument	Camera Height (m)	Camera Tilt (°)	Video Length (min)	Frame Rate (fps)	Pixel Size (m)	Reprojection Error (at Distance)
Duck	Argus	43	68–82	17	2	5	<1 m (<500 m)
Porthtowan	Argus	44	75–85	17	2	5	-
Scheveningen	UAV	110	61	9	2	2	~ 1 m (<200 m) ~ 7 m (400–600 m)
Narrabeen	UAV	89	73	9	2	2	<1 m (<250 m) ~ 7 m (1.5–2 km)

Table 2. Hydrodynamic conditions during field recordings.

Field Site	H_s (m)	T_p (s)	WL (m)
Duck	0.79	5.0	0.08
Porthtowan	1.03	10.0	−0.96
Scheveningen	0.75	5.5	0.60
Narrabeen	1.63	8.5	0.67

In Section 4, the results of the mapping algorithm are presented. The field-site of Duck, NC, USA was used as the lead case to elucidate the processing steps of the workflow for an arbitrary real case. The final results are presented for all field-sites: Duck (NC, USA), Narrabeen (Australia), Scheveningen (the Netherlands), and Porthtowan (UK). The quality of the final results for the depth estimates, d , was assessed by comparison with in-situ bathymetry data. Maps for wave direction and celerity and near-surface currents are discussed in Section 5.

4. Results

The field site of Duck gives an illustrative example (Figure 6) of the processing steps described in Section 3. After decomposing image sequences of Duck (Figure 6a) into 16 dynamic modes, the modes were normalized to global one-component phase images (GOCPI) and filtered for frequency and resolution (as of Figure 4). The frequency filter discarded eight GOCPI, and another four GOCPI failed the criterion of minimum $8 \text{ px}/L_{off}$, leaving four GOCPI for further analysis (Figure 6b). These remaining GOCPI reveal intricate wave patterns and finely capture wave refraction towards the coast. The associated frequencies ω^\dagger (rad/s) $\approx \{0.59, 0.78, 0.97, 1.17\}$ were quite constant across successive image sequences, showing that wave periods $T(\text{s}) = \{10.6, 8.1, 6.5, 5.4\}$ govern the mixed wave field. On the basis of these wave components, the subdomain sizes for local analysis were determined. Stacking them in layers for some grid location revealed a pyramid-shaped cell, and tapering the layers put the focus toward its centre grid point (Figure 6c).

Performing PIV and 2D-FFTs to the autocorrelated layers revealed a sparse spectral point cloud (SSPC) with eight points, with k^\dagger values increasing for increasing ω^\dagger , sketching a typical wave-dispersion curve (Figure 6d). In the example, the PIV and 2D-FFT estimates for k^\dagger were almost indistinguishably close in three of the four spectral layers; however, for the uppermost layer, they lay further apart. In this case, the point that lays off-track corresponds to the 2D-FFT estimate. This is signalled by its quality weight W , which is lower than for the PIV counterpart (Figure 6d, blue vs. yellow point). Augmenting the SSPC with stored SSPCs from the past minute (i.e., from four previous updates) within a radius of $Rad = 75$ m yields the dense spectral point cloud (DSPC) (Figure 6e). The choice of $Rad = 75$ m was arbitrary and represents a balance between extracting information from close by, while capturing sufficient current refraction (especially of shorter period waves). Other choices for Rad may be made, but the algorithm is not too sensitive to this parameter since the surrounding SSPCs within Rad resemble just 25% of the total spectral weight. As desired, this lower weighting is apparent from the corresponding spectral points in the DSPC (Figure 6e, blue points). Note that in contrast to Duck, points in DSPCs of Porthtowan, Scheveningen, and Narrabeen were more dispersed over the frequency domain, because GOCPI frequencies vary more across successive updates (not shown). Finally, the DSPC was fitted with the Doppler-shifted linear dispersion relationship (Equation (1)) to produce characteristic cones corresponding to certain d, \mathbf{U} estimates (Figure 6f, magenta cone). Combining the local estimates from all grid locations yields global maps of d, \mathbf{U} .

These resulting maps were not only quickly retrieved, but they also show that estimates of d are accurate (Figure 7). For all four field sites, depth maps compare with the ground truth. Recall that the algorithm was not tuned to the individual field sites. While the first depth update—after 32 s of video—is still rough, it already gives a clear overall picture of the bathymetry with shallower and deeper regions (Figure 7, 1st update).

Nearshore sandbars at Duck (Figure 7a) are readily visible. With the fifth update—after 96 s of video—(Figure 7, 5th update) the depth maps approximated the ground truth (cf. Figure 7, 5th update and *ground truth*). Estimates quickly improved after the first update, indicating that the temporary spectral storage together with Kalman filtering effectively converged d estimates. Mapping updates continued to improve and become spatially more coherent towards the end of the videos (Figure 7, last update).

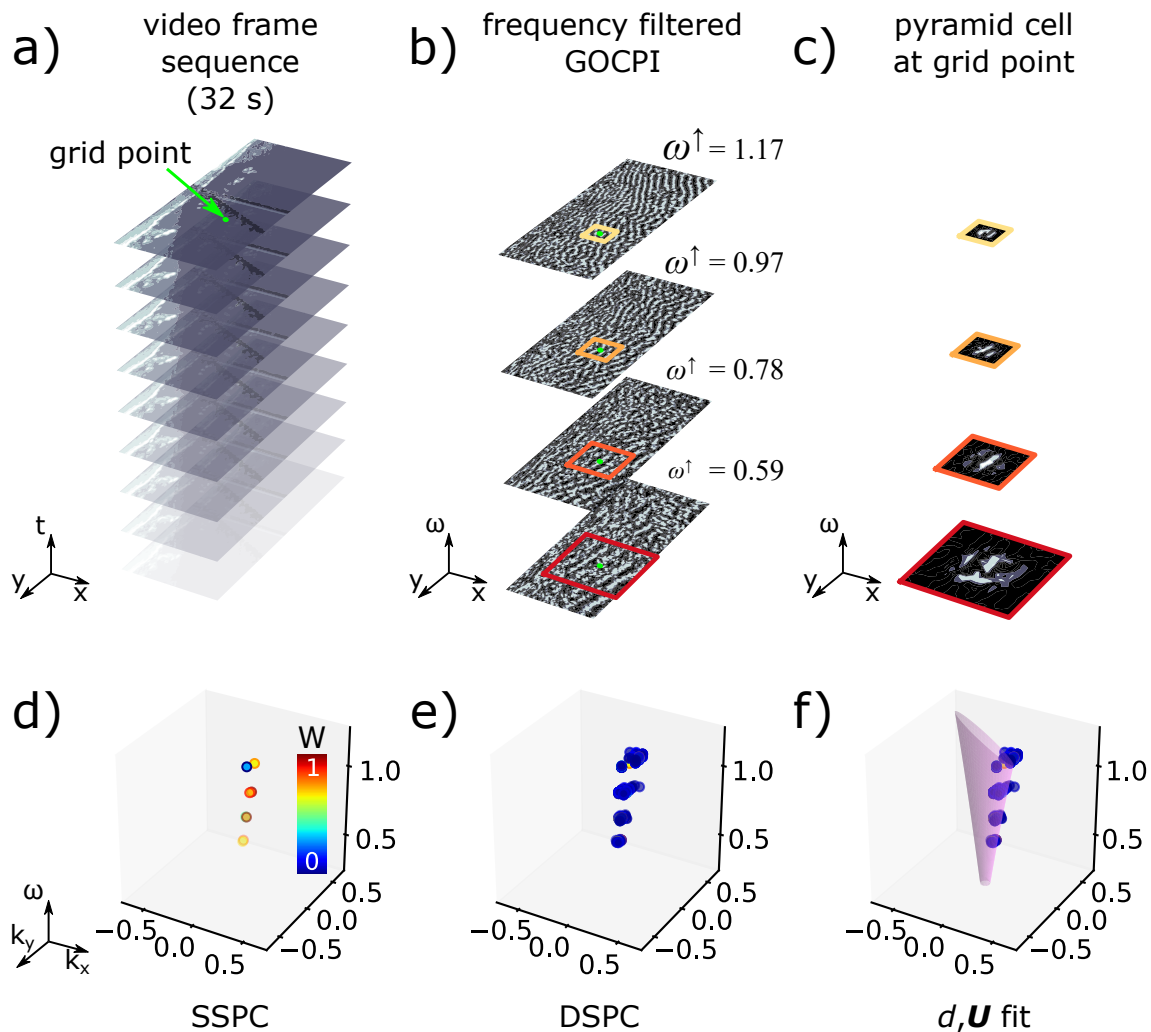


Figure 6. Processing results for a selected grid location at the Duck field site at the 7th update. (a) Image sequences (x, y, t space, see Figure 2 top) used for successive updates depict 32 s of wave movement and have a 50% frame overlap in time. (b) Frequency-filtered global one-component phase images (GOCPI) from dynamic mode decomposition (DMD) (x, y, ω space, see Figure 2 right pathway) uncovering frequencies $\omega^\uparrow = \{0.59, 0.78, 0.97, 1.17\}$ as essential components in the wave field recording. GOCPI outside the gravity wave frequency band are discarded as well as higher frequency GOCPI where 5 m pixel resolution is insufficient to guarantee at least 8 points per offshore wave length (cf. Figure 4, (2) and (3)). (c) The pyramid cell at the grid location, with subdomain layers subsampled for computational speed and tapered with Hanning windows to focus wave information. A colour gradient from red to yellow highlights decreasing subdomain size for increasing frequency. (d) Sparse spectral point cloud (SSPC) (k_x, k_y, ω space; see also Figure 2 bottom), consisting of pairs of \mathbf{k}^\uparrow estimates from FFT and PIV per frequency layer ω^\uparrow . Colours indicate the weight of each estimate (colour scale) (cf. Figure 4, (6)). (e) The SSPC augmented to a dense spectral point cloud (DSPC) using stored spectral data of the grid location and surrounding grid locations within a radius of 75 m (cf. Figure 4, (10)). Blue colours elucidate the lower weighting of stored spectral data. (f) The d, \mathbf{U} fit on the DSPC (magenta cone) using Equations (1) and (2) (cf. Figure 4, (12)).

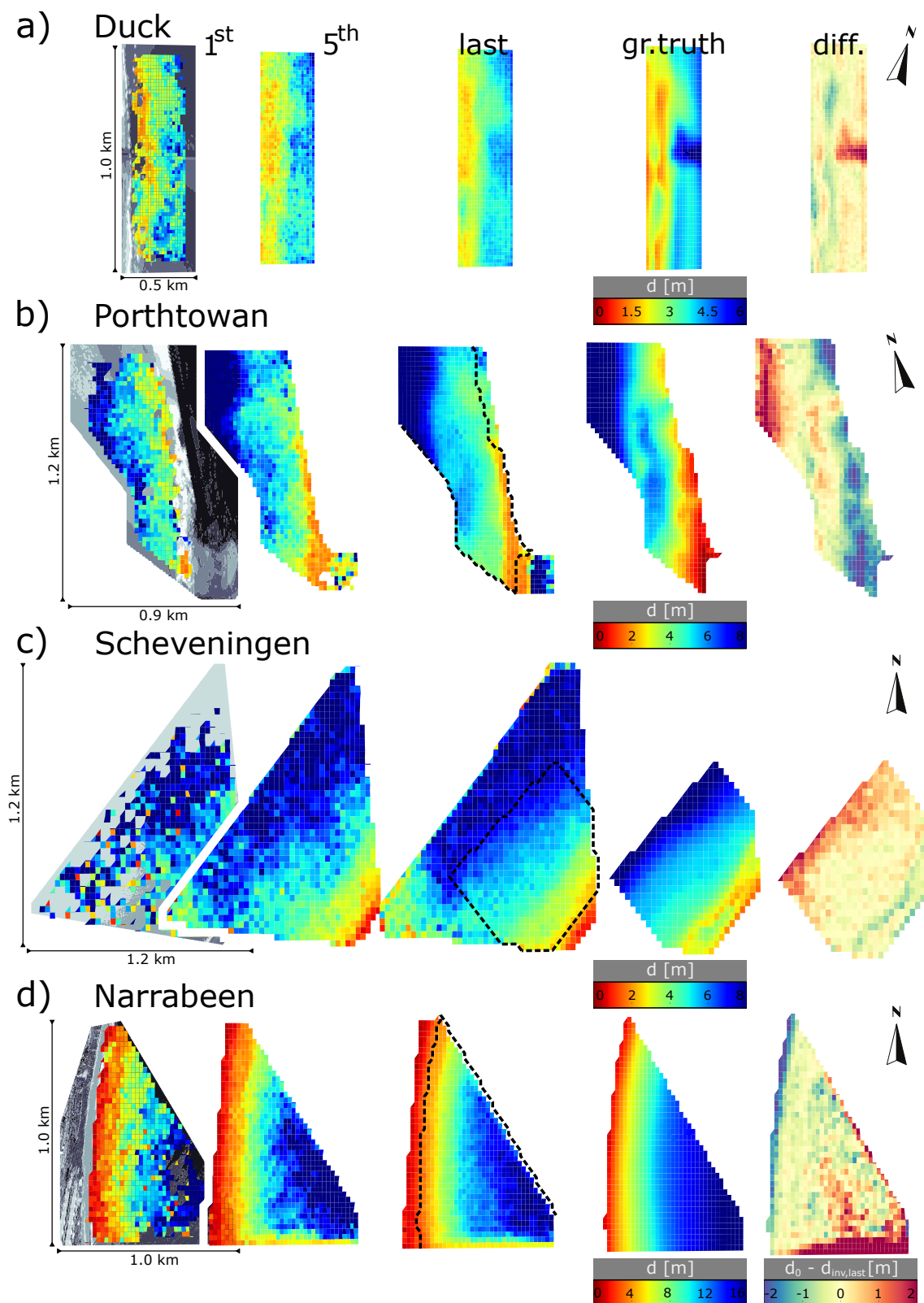


Figure 7. Depth updates from video of the field sites (a) Duck, (b) Porthtowan, (c) Scheveningen, and (d) Narrabeen. In (a–d): left-most panel depicts an example frame (grey scale) of the video with corresponding dimensions; inverted depths (d_{inv}) of the 1st update are overlaid. Following two panels to the right present inverted depths of the 5th and last update. Ground truth measurements ($gr. truth$, d_0) are mapped in the second panel from the right (d_{inv} and d_0 in identical colour scale); the extents are indicated by dashed black lines in panels of last update. Differences between ground truth and last update ($diff.$, $d_0 - d_{inv,last}$) are depicted in right-most panel, with red/blue indicating under/overestimation (colour scale).

For all videos, differences with the ground truth are minimal over large parts of the observed area (Figure 7, *difference*, light areas). Regions with errors $|\Delta d| > 0.5$ m are mostly found in shallow water or the deep water boundary of the observed domain. At Duck and Scheveningen, depth overestimation of $\Delta d \approx -0.5$ to -1.0 m occurs around the sandbars (Figure 7a,c, *difference*, blue areas), while in other shallow parts, the errors are smaller. At Porthtowan and Narrabeen, the overestimations are larger $\Delta d \approx -0.5$ to -2.0 m and generally occur in shallow water (Figure 7b,d, *difference*, blue areas). The reason is probably that wave heights at Porthtowan and Narrabeen are larger $H_s = 1.0 - 1.6$ m (Table 2) and more nonlinear close to shore, compared to Duck and Scheveningen where $H_s < 0.8$ m and wave breaking restricts to the sandbars, after which most wave nonlinearity is lost. Another, physical reason for near-shore depth estimates appearing larger is local wave set-up [26], which is not accounted for in the comparison with the ground truth. Regions and reasons for depth underestimation differ per site. At Duck, depths were only underestimated around the pier, which blocks the view to the underlying waves (Figure 7a, red patch). At Porthtowan and Scheveningen, depths were underestimated by $\Delta d \approx 0.5 - 1.5$ m near the offshore boundary (Figure 7b,c, red patches). At Porthtowan (Figure 7b), the underestimation was mainly caused by the fact that relevant lower-frequency cell layers in near-boundary pyramid cells were too large to be used. The underestimated region also lay >800 m from the camera, where inaccuracies in geometry influence depth estimates [91]. The size of errors in the video geometries (Table 1) suggests a limited effect on depth estimates in general, yet some depth error may be induced in regions further afield (e.g., for $T = 8$ s and $d = 10$ m, an error $\Delta L = 7$ m in wave length causes a depth error of $\Delta d \approx 2$ m; see also Figure 1 of [39]). At Scheveningen (Figure 7b), the underestimated region begins further away from the offshore boundary, where boundary effects should be less pronounced. Here, the underestimation likely stems from the relatively small waves, $H_s = 0.75$ m and $T_p = 5.5$ s, who feel little of the underlying bottom. At Narrabeen, underestimation mainly occurs at the boundary farthest from the camera. This underestimation is likely caused by similar boundary effects as at Porthtowan, since here the underestimated region also lies more than 900 m from the camera. In conclusion, depth maps can show regions that are less accurate, yet all in all, the maps approximate local bathymetries: on average 80% of the mapped area had errors $\Delta d < 1$ m.

Direct comparison of estimated depths against the ground truth confirms generally accurate depth maps (Figure 8). Median depth estimates per given depth are mostly close to ground truth (cf. Figure 8, double green curves and black 1:1 line). For the video of Duck (Figure 8a), the similarity is visible for the entire depth range from $d = 1.5 - 5.0$ m, except for the scour hole where the pier obscures the underlying waves. At Porthtowan, median depth estimates deviate more from the ground truth but remain $\Delta d < 2.0$ m for depths $d < 10$ m (Figure 8b). Errors are largest in the breaking region where depths are $d = 2 - 3$ m, which is a common observation in depth inversion studies e.g., [92]. This is similar for Narrabeen (Figure 8d). Here, however, differences between median depth estimates and the ground truth are minimal for the entire depth range beyond that, $d = 3 - 17$ m. In contrast to Porthtowan ($H_s = 1.0$ m), the wave heights at Narrabeen were larger ($H_s = 1.6$ m); moreover, the few large boundary errors in the statistics were outnumbered by otherwise accurate estimates (Figure 8d, relatively few outliers). Median depth estimates at Scheveningen (Figure 8c) also confirm earlier spatial observations of slight depth overestimation near the shallow sandbar and underestimation for $d > 7$ m.

Although for each field site the first mapping update after 32 s of video was considerably more scattered compared to the last update (cf. Figure 8, white and blue scattered dots), the median estimate is comparable (cf. Figure 8, double grey and double green curves). Hence, the first mapping updates already give a rudimentary impression of the bathymetry, albeit with more local uncertainty. It suggests that waiting for many mapping updates is superfluous and that it may be efficient to stop after a couple of updates, when a certain degree of accuracy in the depth map is reached.

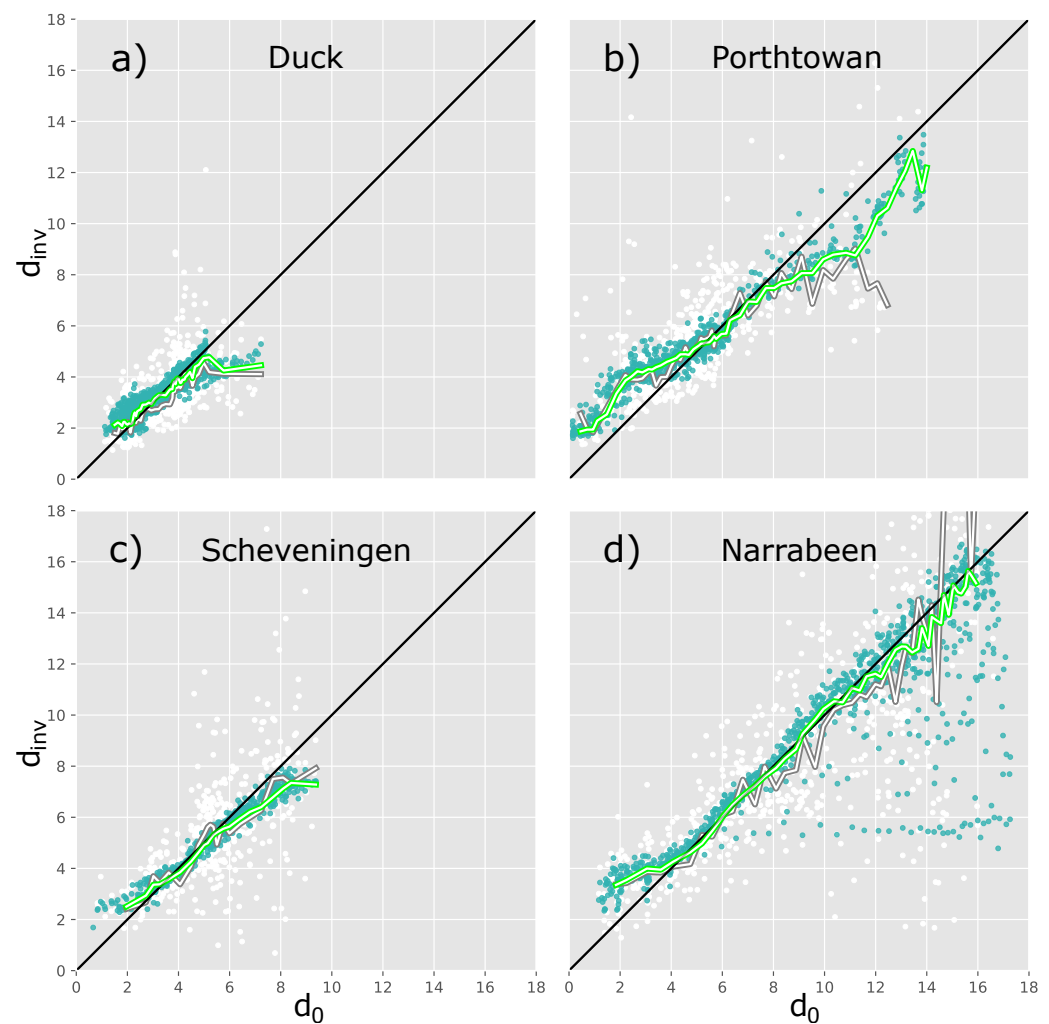


Figure 8. Direct comparison between inverted depths (d_{inv}) and ground truth depths (d_0) for field sites (a) Duck, (b) Porthtowan, (c) Scheveningen, and (d) Narrabeen. Coloured dots correspond to the last update whose median is shown with a double green line. Analogously, underlying white dots correspond to the first update whose median is shown with a double grey line. Potential water-level differences within the observed domains, due, for example, to near-shore wave set-up, were not accounted for in the comparisons.

Bulk errors decreased with the increasing number of mapping updates (Figure 9). Since depth errors are not always constant and linearly distributed over depth (Figure 8b) and may contain outliers (Figure 8d), the median bias was adopted to quantify structural over- or underestimation, and the interquartile range (IQR = 75th – 25th percentile) was used to measure the scatter. In comparison to other common error measures, note that Duck, with quite linear and normally distributed depth errors (see Figure 8a), had a median bias that was almost identical to the commonly used mean bias, and its IQR was close to the root mean square error (not shown). Absolute median biases (Figure 9, dashed lines) start small $|\Delta d_{bias}| < 0.5$ m with the first update and eventually become $|\Delta d_{bias}| < 0.1$ m at Duck (with update 3), Scheveningen (with update 27), and Narrabeen (with update 10). The median bias at Porthtowan reached $\Delta d_{bias} \approx -0.25$ m. The IQRs (Figure 9, solid lines) decreased fast over the first couple of updates, signalling fast improvements in the accuracy of the depth map. After that, the convergence rates started to relax. The total IQR improvements were 0.9 m \rightarrow 0.5 m (Duck), 1.7 m \rightarrow 1.4 m (Porthtowan), 1.6 m \rightarrow 0.6 m (Scheveningen), and 2.6 m \rightarrow 1.1 m (Narrabeen). The elbow in the exponentially decreasing curves represents a compromise between the size of error and the number of updates. It is difficult to pinpoint the exact locations of the elbows, but for Duck and Porthtowan,

they appeared to be somewhere between the 2nd–6th update, while for Scheveningen and Narrabeen they were more likely located between the 5th–10th update. The reason for the different elbow positions probably roots in the instrumentation. The videos of Duck and Porthtowan were recorded with stationary Argus stations, while the videos of Scheveningen and Narrabeen were recorded with UAVs. In contrast to the Argus stations, UAVs have some freedom to move and are not professionally tuned to the field site, which likely causes depth errors to be initially large; however, these errors decrease rapidly. Concluding the previous findings, a general rule of thumb might be to stop with the 5th update, after 1.5 min of video. Awaiting more updates may further improve results but also requires more computation time and a larger video length. A single estimate based on 32 s of video is enough to get a rough depth map. In the end, the use of the data, and desired accuracy, determine whether to stop the analysis sooner or later.

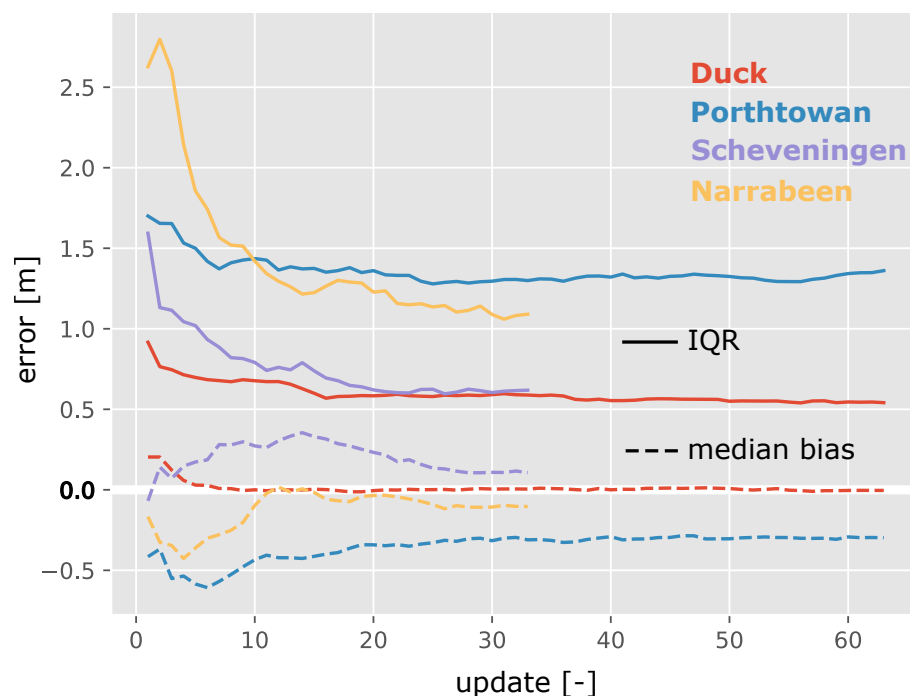


Figure 9. Depth errors at successive updates for field sites Duck (red), Porthtowan (blue), Scheveningen (purple), and Narrabeen (orange). Dashed lines present a median bias. Solid lines present confidence intervals by the interquartile range (IQR = 75th–25th percentile).

5. Discussion

Alongside the maps of depth, mapping updates for hydrodynamics, wave celerity, \mathbf{c} , and local near-surface currents, \mathbf{U} , were retrieved. Since no ground truth data are available for these parameters, they are discussed qualitatively in Section 5.1. The subsequent Section 5.2 discusses the algorithm's ability to process video on-the-fly.

5.1. Maps of \mathbf{c} and \mathbf{U}

While vector fields of \mathbf{c} remain quite stable after the first mapping update, vector fields of \mathbf{U} require more updates to converge. Here, approximately the 10th update (Figure 10) shows convergence of the local current patterns. Additionally, the current velocities and directions appear realistic. Note that local changes in \mathbf{c} and \mathbf{U} still occur after the 10th update, as allowed by the process variance in the Kalman filter (not shown).

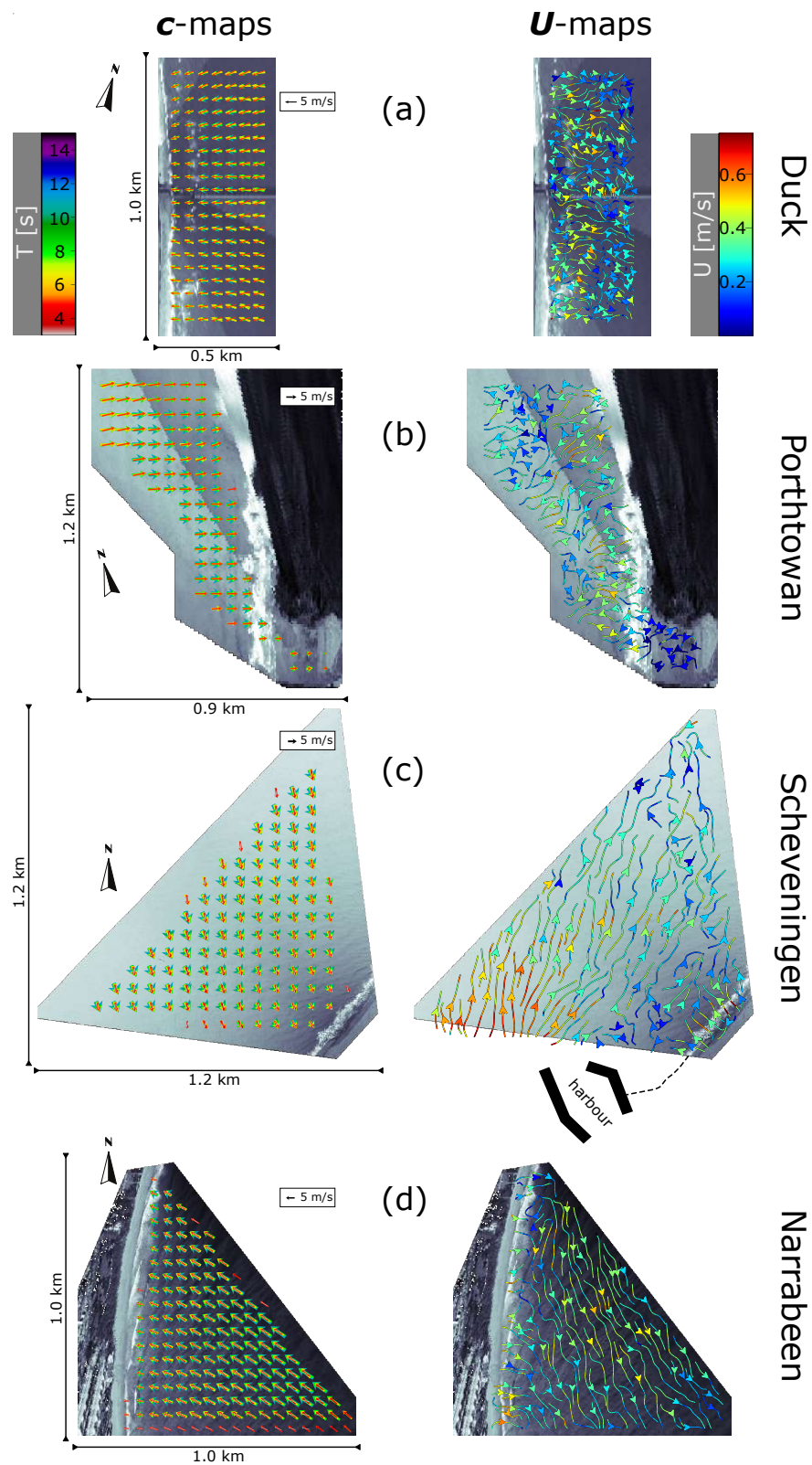


Figure 10. Maps for wave celerity, c , and near surface currents, U , for field sites (a) Duck, (b) Porthtownan, (c) Scheveningen, and (d) Narrabeen. The maps exemplify the 10th update. Wave celerity vectors of different wave periods, T , are superimposed and coloured according to top-left colour scale. Near-surface currents are indicated by streamlines whose colours highlight the current magnitudes as of top-right colour scale. In the U -map of (c), the Scheveningen harbour is outlined in black.

Maps of c reflect the general direction of wave propagation for all videos (Figure 10, left column). Local vectors are oriented normally to the recorded wave crests and trace characteristic wave refraction towards the coastline. Mean wave directions per wave period are quite similar for all field sites, except for Duck, which shows some directional spread between shorter and longer period waves (Figure 10a left, red vs. yellow arrows). Wave celerity decreases towards shallower water, being most apparent for Porthtowan and Narrabeen. This was expected as cross-shore differences in depth were the largest at these locations (Figure 10b,d, left). Moreover, the wave celerity of long-period waves started to decrease further offshore than for short-period waves (Figure 10d left, starting at offshore boundary, green arrows already get shorter while red arrows had constant length until close to the shoreline). This was also expected since longer waves feel relatively smaller depths.

Maps of U show intricate patterns that are largely consistent in time and space (Figure 10 right). Coastal currents are typically tide or wave driven [85] and have magnitudes of decimetres per second (dm/s), e.g., [93]. They can reach meters per second (e.g., strong rip currents [94] such as the “Backpackers’ Express” at Bondi Beach [95]), but such conditions are not likely at the field sites analysed in this study. With mostly 0–5 dm/s, the maps of U had the correct order of magnitude. At Duck, near-surface currents close to shore often pointed in offshore directions, while further from shore they also pointed in southerly directions (Figure 10a right). The near-shore offshore-directed flows probably represent the effect of undertow caused by normally incident waves (see Figure 10a left). It is interesting to note that surface current estimates using optical flow on wave-averaged images might predict opposite flow directions in shallow water, namely, shoreward directed currents, as recent data analyses for a similar wave situation at Duck suggest [96]. The hypothesis is that current estimates from optical flow are more indicative of Stokes drift, while wave-inverted current estimates capture the undertow. Especially around the thinner part of the sandbar in the south (see also Figure 7a, *ground truth*), streamlines from northerly and southerly directions often converge to form a common flow direction offshore, which could be indicative of a local rip current and which is not uncommon at this site [97]. Although it is visible throughout many updates (not shown), it is often hard to recognize among enlarged U estimates on the sandbars. Note that U estimates on the sandbars are influenced by distorted wave celerity estimates of breakers, which simultaneously leads to larger depth errors (see Figure 7a, *difference*). At Porthtowan, near-surface current estimates generally point in south and south-easterly directions, except for the deeper region where directions are less coherent. Additionally, at shallow parts, currents point away from the coast (Figure 10b right). The southerly directed flows may reflect some remaining tidal ebb flow consistent with the time of the video recording approaching low water. Similar to Duck, offshore-directed flows close to the shoreline are suggestive of a cross-shore-directed undertow under almost shore-normal wave incidence. Near-surface current estimates at Scheveningen were well in line with expectation. The video recording was taken just above a harbour (Figure 10c right). During flooding, the longshore tidal current accelerates around the harbour jetties forcing larger current velocities, and a characteristic region with eddies and opposite directed flows is formed on the lee side of the northern jetty. Both effects were visible, and current magnitudes of mostly 0.4–0.6 m/s off the coast also agree with typical current magnitudes off the nearby Rotterdam coast. Close to the coastline, U estimates are again offshore-oriented, suggesting undertow. Similar to Scheveningen, coastal currents at Narrabeen run mostly alongshore. Here, however, they are likely not driven by tides but by waves instead. The direction of flow at this location is sensitive to the angle of wave incidence and alongshore differences in wave height and dissipation, e.g., [98]. For the recorded situation, the angle of wave incidence suggests northward flow; however, alongshore differences in wave height may still force a southward flow as mapped. The true current direction remains uncertain at this site. Summarizing, near-surface current estimates are coherent and can be explained, but the Scheveningen field-site is most relatable to expected tidal flow patterns (Figure 10c, right).

5.2. On-The-Fly Processing

Maps of d , c , and U are realistic and ideally returned on-the-fly. To allow for an on-the-fly analysis during this study, specific choices were made to balance calculation time and spatiotemporal resolution of mapping updates. The algorithm marches forward in time by consecutively analysing small sequences of video frames, giving mapping updates after each sequence. The frame sequences partly overlap (see legend Figure 11a). For pre-recorded video, the overlap could be controlled. It was 50%, since frame sequences were 32 s long, and the algorithm was set to march forward in 16 s intervals. For on-the-fly analysis of a video feed, this approach does not work, as processing times start to play a role. A mapping update now needs to reflect the current situation and therefore is mostly based on the latest recorded video frames. Say the processing time of a frame sequence was 20 s, then during the processing, 20 s of new video frames have been recorded. The next frame sequence to analyse includes these new 20 s plus 12 s from the previous frame sequence representing the overlap. The amount of overlap hence depends on the processing time, which may vary per update. Statistically, the shorter a frame sequence, the more it represents a random sample of the wave field, where waves vary in height and direction, currents change, etc. [66]. Additionally, the video quality itself may vary due to changes in lighting, the accuracy of the orthorectification, etc. This impacts the processing time, which may hence be slower or faster causing smaller or larger overlap, respectively, between consecutive frame sequences.

For the results presented up until now, the processing time for an update—using a standard laptop with four CPUs and a working memory of 4 GB—was typically 30–60 s. If the fixed, 16 s overlap was changed to instead be variable-based on the processing time, this would be too slow. To enter the realm of on-the-fly computation, the processing time hence needs to be further reduced. The three main elements influencing processing time are: (i) the computational power, (ii) the grid resolution, and (iii) the amount of spectral data points stored and used to locally derive c , d , U . In this study, the effect of (ii) and (iii) was considered using the video of Duck as a testcase:

(ii) In a first step to increase computational speed, the grid resolution was reduced from 1260 to 720 grid points. As required to simulate on-the-fly analysis, the overlap was set to depend on the processing time. The resulting simulation returns mapping updates in variable time intervals with correspondingly variable overlap between analysed frame sequences (Figure 11a). Yet, on average, every 23 s, a mapping update is given. Although the grid resolution is lower, the bathymetry is still estimated in reasonable detail, as well as wave propagation and near-surface currents. At the 5th update, after 102 seconds, the results again converged to a large extent. The analysis could at this point be deemed on-the-fly; however, more frequent updates would further improve the user experience. A detailed view of the first ≈ 4 updates reveals that they were returned at a faster pace than consecutive updates (Figure 11a, orange lines closer together, and more, darker green overlap between frame sequences). During the first updates, the amount of spectral data in storage still increased towards full capacity, which suggests that the rate of updates should increase for a smaller number of spectral data points.

(iii) Reducing the number of spectral data points that are stored and used reduces the number of spectral points in the dense spectral point clouds (DSPCs). This should be acceptable as long as the true local wave spectra are well represented by the DSPCs. By construction, a DSPC represents an aggregate of (mostly stored) sparse spectral points clouds (SSPCs) (Section 2). These SSPCs stem from surrounding grid points within a radius Rad (Appendix C, Table A2, $Rad = 75$ m) from the location under analysis and from previous updates within a short period where the wave signal is assumed to be stationary (see Section 2 and Figure 4). Now instead of using SSPCs from all grid points within Rad , a random subset can be selected. Here, a random subset of 12 surrounding grid points was selected from the full set of 48 grid points. The amount of SSPCs from this subset of grid points can be reduced even further by simply storing fewer SSPCs. The storage time, which is the time that the wave signal is assumed stationary (Section 2), was here

reduced from 60 s to 30 s. The effect of both measures on processing speed was significant, while mapping results remained similar (not shown). The 5th update now occurred approx. 20 s earlier and the 10th update even 80 s earlier compared to the simulation with only a reduced grid resolution (cf. Figure 11a,b). Updates were, on average, returned every 13 s, which may be considered a quite continuous temporal output, as desired for on-the-fly application (Figure 11b). It is noteworthy to mention that convergence rates appear to depend rather on the number of updates than on the timing (not shown), which suggests that it may be favourable to return updates at a faster pace; however, a detailed analysis remains a subject for future study.

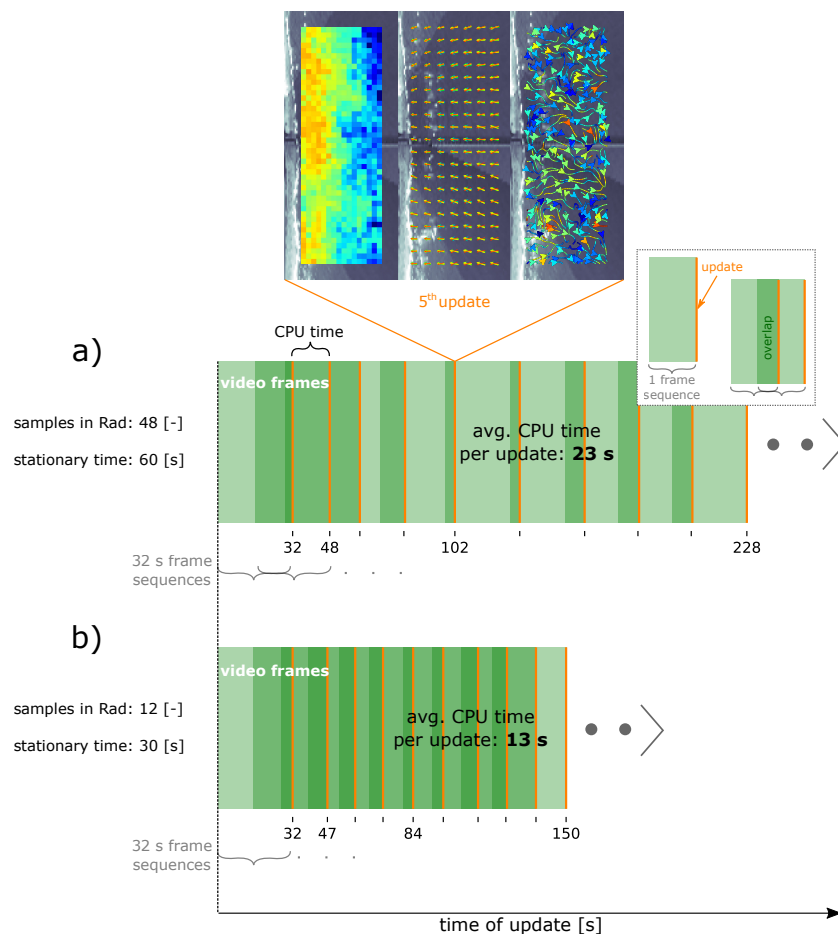


Figure 11. On-the-fly analysis of Duck video. Video frame sequences of 32 s (green boxes, see legend) were consecutively analysed with mapping updates after each sequence (vertical orange lines). Depending on the computational processing time (CPU time), frame sequences overlapped more or less (darker green, see legend). The CPU time mainly depends on the used machine, the grid resolution, and the amount of spectral data. The amount of spectral data is controlled by the number of sample grid points within a radius (*Rad*; see Section 2 and Figure 4) from each location and the duration that spectral data are stored and used, which is the duration the wave signal is assumed stationary. Timings of the first ten updates are shown for (a) 48 samples in *Rad*, stationary time 60 s and (b) 12 random samples in *Rad*, stationary time 30 s. Both grids consist of 720 grid cells. For (a), the 5th mapping update is visualized as an example.

6. Conclusions

This study describes a fast and self-adaptive algorithm to map coastal parameters on-the-fly from aerial wave imagery. Updates of depth, d , wave propagation, c , and near-surface currents, \mathbf{U} , are returned every few seconds, such that a video feed can theoretically be processed on-the-fly, and a user does not need to wait and engage in post-processing. The input requires orthorectified video with known pixel size (m) and frame rate (fps).

Apart from that, the algorithm works unsupervised for the presented field sites. The basis for fast computational speed and increased automation lays in the use of the dynamic mode decomposition (DMD), which is a dimensionality reduction technique that disposes redundant video information. It reduces the video to a set of, here, 16, intrinsic wave patterns and autonomously finds the corresponding frequencies. Applying a DMD, the search for optimal spatial sampling schemes can also be automated, such that in the end, no manual choices underlie the local wave spectra, which are the basis to derive c and invert d , \mathbf{U} . Consecutive mapping updates are improved by taking specific measures, such as using an innovative system to temporarily store and call up spectral data but also by penalizing spectral outliers through a loss-function and Kalman filtering. The algorithm's potential for all-round application was demonstrated using video data from stationary camera installations and UAVs, recorded at four different field sites, Duck (USA), Porthtowan (UK), Scheveningen (NL), and Narrabeen (AU), with hydrodynamic conditions ranging between $H_s = 0.75 - 1.63$ m and $T_p = 5 - 10$ s. Validating the depth maps, d , showed that they accurately reflected ground truth measurements. The maps quickly improved from the first update to typically the fifth update, at 1.5 min into the video, after which the rate of improvement relaxed. It suggests that for time-efficient coastal mapping of depths, 1.5 min of video suffice, such that the next location of interest can be recorded. The interquartile range (IQR) of depth errors decreased from the first update, with minimum and maximum values of 0.9 m (Duck) and 2.6 m (Narrabeen), respectively, to the last update, with values of 0.5 m (Duck) and 1.3 m (Porthtowan). Absolute depth biases were small throughout all updates, slightly improving from minimum and maximum values of respectively 0.1 m (Scheveningen) and 0.4 m (Porthtowan) at the first update, to 0.0 m (Duck) and 0.3 m (Porthtowan) at the last. Maps of wave celerity, c , and near-surface currents, \mathbf{U} , could not be validated at this stage; however, qualitative assessment showed vector fields of c to match observed wave propagation and vector fields of \mathbf{U} to match expected tide- and wave-induced currents. The algorithm finally demonstrated its potential for on-the-fly video feed analysis by taking computational processing times into account. Therewith the groundwork is laid for a fast and easy-to-use tool for coastal reconnaissance. Striving towards universal applicability, more field cases and community-driven development are desired, wherefore the code and data are openly available at <https://doi.org/10.4121/c.5704333> (accessed on 22 November 2021).

Author Contributions: Conceptualization, M.G. and S.d.V.; methodology, M.G.; software, M.G.; validation, M.G.; formal analysis, M.G.; investigation, M.G.; resources, M.G. and S.d.V.; data curation, M.G.; writing—original draft preparation, M.G.; writing—review and editing, M.G., S.d.V., and S.A.; visualization, M.G.; supervision, S.d.V. and S.A.; project administration, S.d.V. and S.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the TU Delft and COCOS-Lab project as part of the Stichting ZABAWAS research program.

Data Availability Statement: The original code and data presented in this study are openly available in 4TU Research Data at <https://doi.org/10.4121/c.5704333> (accessed on 22 November 2021). The code can also be found at <https://github.com/MatthijsGawehn/COCOS> (accessed on 22 November 2021).

Acknowledgments: We are grateful to all the people that provided data for this article. For the data of Duck, we would like to thank Katherine Brodie, Margaret Palmsten, Rob Holman, the Coastal Imaging Lab at Oregon State University, and the U.S. Army Engineer Research and Development Center's Field Research Facility. Data of Porthtowan have been collected and kindly shared by Erwin Bergsma and the Coastal Processes Research Group at the University of Plymouth. We would like to thank Roeland de Zeeuw, Sean Zandbergen, Janbert Aarnink, and Shore Monitoring and Research in the Hague for collecting and sharing the data of Scheveningen. Video data of Narrabeen have been collected and kindly provided by Kilian Vos, Chris Drummond, Kristen Splinter, and Mitch Harley and the corresponding bathymetry validation data by the New South Wales Department of Planning, Industry, and Environment. Thanks to Nathan Kutz and Steve Brunton from the University of Washington for openly sharing great lectures and code on the dynamic mode decomposition.

Special credit goes to Grant Sanderson for insightful discourses in mathematics on his YouTube channel 3blue1brown.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Review of Strategies to Extract Spectral Gravity Wave Signatures

- *Dugan et al. [11], Young et al. [20], Ziemer [21], Irani et al. [37]*

The most straightforward strategy is to transform local video cut-outs from 3D \mathbf{x}, t into 3D \mathbf{k}, ω spectra via 3D fast Fourier transforms (3D-FFTs). An energy threshold can then be used to separate the spectral footprint of gravity waves $\mathbf{k}^\uparrow, \omega^\uparrow$ from the noise floor. A benefit of this approach is the possibility to retrieve spectral data up to two times the Nyquist frequency [18], which can be important if frame rates are low (e.g., slow radar rotation speeds).

- *Senet et al. [30]*

The strategy starts as [11,20,21,37], but with reduced size of the video cut-out, which requires a follow-up step: after producing 3D \mathbf{k}, ω spectra, they are sliced into separate 2D wave-number layers \mathbf{k}_j, ω_j per constant frequency ω_j . These spectral layers are then filtered to separate the wave-number signature of gravity waves from the noise floor. Subsequently, using inverse 2D fast Fourier transforms (2D-FFT⁻¹), each filtered 2D \mathbf{k}_j, ω_j layer is transformed back to the spatial domain to produce a corresponding complex-valued image (\mathbf{x}_j, ω_j) , which depicts the local wave pattern associated to ω_j . Being associated to one frequency component and neglecting spatial differences in amplitude (which can be achieved through entry wise normalization), the complex valued images represent LOCPI. Finally, assuming the wave field is homogenous, spatial gradients in each LOCPI yield a local representative vector \mathbf{k}_j^\uparrow associated to that frequency component $\omega_j = \omega_j^\uparrow$. The suggested benefit of extracting \mathbf{k}_j^\uparrow from LOCPI, instead of straight from 3D \mathbf{k}, ω spectra, is that the reduced size of the video cut-out leads to better localisation of $\mathbf{k}^\uparrow, \omega^\uparrow$.

- *Stockdon and Holman [39], Holman et al. [43], Plant et al. [99]*

Similar to [30], this strategy constructs one-component phase images. First, the video is transformed from 3D \mathbf{x}, t to 3D \mathbf{x}, ω via FFTs in time. This process finds GOCPI per Fourier frequency but without taking spatial coherence into account. A subsequent local analysis aims to find this spatial coherency. Cross-spectral matrices are computed for predefined frequency bands with central frequencies ω_c . The idea is that the strongest eigenvector of each cross-spectral matrix, after entry-wise normalization to unit magnitude, resembles a spatially coherent LOCPI (\mathbf{x}_c, ω_c) corresponding to ω_c . Analogous to [30], \mathbf{k}^\uparrow is finally deduced from phase gradients and $\omega_c = \omega_c^\uparrow$. A benefit of this strategy is that the video can be sampled in any fashion (e.g., non-regularly) in preparation of local cross-spectral matrices. A drawback of this strategy is the uncertainty around the frequency associated to \mathbf{k}^\uparrow as it lies within a frequency band and needs to be approximated by the bands' central frequency ω_c^\uparrow .

- *Simarro et al. [53]*

The most recently developed strategy suggests that globally coherent GOCPI can be found through a singular value decomposition of the time-analytic signal of the video. First, the video is reshaped into a matrix \mathbf{X} , whose columns represent the video frames squeezed into arrays and whose rows hence contain the timeseries of a pixel. Each timeseries is converted into its analytic signal using the Hilbert transform, which makes the timeseries complex valued (with a single-sided frequency spectrum) and

prepares for a natural retrieval of phases. A singular value decomposition (svd) of the video matrix \mathbf{X} , Equation (A1), then describes the video as a sum of modes, given by pairs of spatial structures and their associated temporal evolution:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \sum_j \overbrace{\sigma_j \mathbf{u}_j \mathbf{v}_j^*}^{\text{jth mode}}, \quad (\text{A1})$$

where the orthonormal columns \mathbf{u}_j of \mathbf{U} represent the (squeezed) spatial structures ($\neq \mathbf{U}$ of Equation (1)), the columns \mathbf{v}_j of \mathbf{V} the associated temporal evolution, and the diagonal matrix contains the singular values σ_j , which sorted in decreasing order denote the contribution of each mode to the total variance in \mathbf{X} . The asterisk denotes the complex conjugate transpose.

Unfolding \mathbf{u}_j into the two spatial video dimensions now provides GOCPI. Simarro et al. [53] argue that the associated time evolution \mathbf{v}_j closely corresponds to a fixed-frequency oscillation ω_j and can therefore be estimated by the averaged temporal phase gradient. This moreover implies that \mathbf{u}_j *approximately* resembles global *one-component* phase images (\mathbf{x}_j, ω_j) , that is, GOCPI. In practice, the final deduction of local \mathbf{k}^\uparrow is done from phase gradients in local subdomains of the GOCPI.

However, the GOCPI being only *approximately* one-component (i.e., $\widetilde{\text{GOCPI}}$, where \sim denotes approximate) flags a deeply rooted issue: using the svd on a wave signal, whether analytic or not, is prone to the mixing of Fourier modes, meaning that wave patterns of different frequencies mix together in \mathbf{u}_j (Equation (A1)) (see [54], Figure 6). The problem can be understood from the fact that the svd result for \mathbf{u}_j is invariant to the ordering of the video frames [54]. As such, these $\widetilde{\text{GOCPI}}$ s do not reflect the distinctly sinusoidal time dynamics of ocean waves.

Table A1. Pros and cons of different strategies used to retrieve local gravity wave $\mathbf{k}^\uparrow, \omega^\uparrow$ signatures from video of a moving wave-field.

Strategy	Pros	Cons
e.g., [37]	<ul style="list-style-type: none"> • Simple approach • Spectral data: $2 \times$ Nyquist limit 	<ul style="list-style-type: none"> • Comp. load: full-dimensional video
[30]	<ul style="list-style-type: none"> • LOCPI: sugg. higher localisation • Spectral data: $2 \times$ Nyquist limit 	<ul style="list-style-type: none"> • comp. load: full-dimensional video
[43]	<ul style="list-style-type: none"> • LOCPI: sugg. higher localisation • Freedom local pixel sampling 	<ul style="list-style-type: none"> • Comp. load: full-dimensional video • approximate ω^\uparrow
[53]	<ul style="list-style-type: none"> • $\widetilde{\text{GOCPI}}$: sugg. higher localisation • Freedom global pixel sampling • Globally coherent wave patterns • Predict subdomain sizes from ω^\uparrow • Comp. load: reduce dimensionality 	<ul style="list-style-type: none"> • Sensitive to mixing Fourier modes
proposed	<ul style="list-style-type: none"> • GOCPI: sugg. higher localisation • Freedom global pixel sampling • Globally coherent wave patterns • Predict subdomain sizes from ω^\uparrow • Comp. load: reduce dimensionality 	

Appendix B. Optimized DMD Based on Variable Projections

Appendix B.1. Synopsis

The optimized DMD by Askham and Kutz [60] finds a solution to Equation A2 via non-linear least-squares minimization (Equation (A3)).

$$\mathbf{X}^T \approx \Psi(\boldsymbol{\omega}, t)\mathbf{B} \quad (\text{A2})$$

or

$$\overbrace{\begin{bmatrix} - & \mathbf{x}_1 & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{x}_m & - \end{bmatrix}}^{m \times n} \approx \overbrace{\begin{bmatrix} e^{\omega_1 t_1} & \dots & e^{\omega_r t_1} \\ \vdots & \ddots & \vdots \\ e^{\omega_1 t_m} & \dots & e^{\omega_r t_m} \end{bmatrix}}^{m \times r} \overbrace{\begin{bmatrix} - & \boldsymbol{\beta}_1 & - \\ \vdots & \vdots & \vdots \\ - & \boldsymbol{\beta}_r & - \end{bmatrix}}^{r \times n}$$

solved via

$$\text{minimize } \frac{1}{2} \|\mathbf{X}^T - \Psi(\boldsymbol{\omega}, t)\mathbf{B}\|^2 \quad (\text{A3})$$

Tailored to the case of processing grey-scaled video frames, the rows $\mathbf{x}_1 \dots \mathbf{x}_m$ of $\mathbf{X}^T \in \mathbb{R}^{m \times n}$ denote the m frames of the video, each squeezed into an array of n pixels (e.g., 10×10 frame becomes $n = 100$ array). Typically, the number of pixels is larger than the number of video frames, $n > m$. The superscript T signifies that the video matrix \mathbf{X} is transposed in this DMD formulation. The complex valued matrix $\Psi(\boldsymbol{\omega}, t) \in \mathbb{C}^{m \times r}$ holds in its columns the timeseries of r sinusoids with frequencies $\boldsymbol{\omega} = \omega_1 \dots \omega_r$ over time $t_1 \dots t_m$ such that $\Psi(\boldsymbol{\omega}, t)_{i,j} = e^{\omega_j t_i}$. The dependency on t_i is implicit (i.e., t_i are known). The rank r is a choice, typically $r \sim O(1) - O(10)$, such that $r < m < n$. The rows $\boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_r$ of $\mathbf{B} \in \mathbb{C}^{r \times n}$ resemble weighted dynamic modes coupled to the frequencies $\omega_1 \dots \omega_r$. The key to “variable projections” [60] is that Equation (A2) can be solved purely by optimizing $\boldsymbol{\omega}$: Say $\boldsymbol{\omega}$ was known, then $\mathbf{B} \approx \Psi(\boldsymbol{\omega}, t)^\dagger \mathbf{X}$ and Equation (A3) becomes $\text{minimize } \frac{1}{2} \|\mathbf{X} - \Psi(\boldsymbol{\omega}, t)\Psi(\boldsymbol{\omega}, t)^\dagger \mathbf{X}\|^2$, which can be iteratively solved using a Levenberg–Marquardt algorithm. For computational details, see [60].

Finding a local minimizer for $\boldsymbol{\omega}$ automatically finds corresponding \mathbf{B} , whose weighted dynamic modes $\boldsymbol{\beta}_j$ reflect pre-products of one-component phase-images. The frequencies and corresponding phase-images are hence found together as the optimal building blocks to form the video matrix \mathbf{X}^T . Note that conforming to the introduction of the DMD in Section 2.1, the weighted dynamic modes $\boldsymbol{\beta}_j$ resemble (scaled) eigenvectors of the linear model \mathbf{A} in the system of differential equations $dx/dt = \mathbf{A}x(t)$. This is shown in Appendix B.2. Weighted dynamic modes $\boldsymbol{\beta}_j$ are split into a dynamic mode with unit norm, $\boldsymbol{\varphi}_j = \boldsymbol{\beta}_j / b_j$ with associated spectral weight $b_j = \|\boldsymbol{\beta}_j\|$. The complex valued entries of $\boldsymbol{\varphi}_j$ differ in magnitude, accounting for spatial differences in the importance of the dynamic mode. To retrieve a phase-image, all entries of $\boldsymbol{\varphi}_j$ need to be normalized to unit magnitude, upon which the array can be reshaped back into a plane of the original video-frame dimensions.

To boost computational speed, the dimensionality of the video matrix \mathbf{X}^T can first be strongly reduced $m \times n \rightarrow m \times r$ by projecting \mathbf{X} onto the first r columns $\mathbf{U}_r = \mathbf{u}_1, \dots, \mathbf{u}_r$ in \mathbf{U} of its svd (Equation (A1)), $\tilde{\mathbf{X}} = \mathbf{U}_r^T \mathbf{X}$. Substituting $\tilde{\mathbf{X}}^T$ for \mathbf{X}^T in Equations (A2) and (A3) yields a set of deflated dynamic modes $\tilde{\mathbf{B}}$, which are straightforwardly inflated back to the original video frame dimensions, via $\mathbf{B}^T = \mathbf{U}\tilde{\mathbf{B}}^T$ (see [60], Algorithm 3). The weights and frequencies stay the same.

The optimized DMD needs proper initialization [60]. The implicit trapezoidal rule (i.e., 2nd-order Adams Moulton formula) in the exact DMD-like Algorithm 4 of Askham and Kutz [60] may fail a proper initialization of the optimized DMD. This especially occurs if $r > 1.5\text{--}2$ times the actual number of components in the data. It is therefore advisable to use a 3rd- or 4th-order Adams–Moulton formula instead ([100], p. 466). For regular, equispaced video frames, alternatively a standard or exact DMD algorithm ([55], Algorithm 1 or Algorithm 2) can also be used as initializer; however, the computation requires more working memory.

Appendix B.2. Eigenvectors \mathbf{B} of Linear Model \mathbf{A}

Askham and Kutz [60] and Boyce and DiPrima [100], pp. 414–419

A system of differential equations $d\mathbf{x}/dt = \mathbf{A}\mathbf{x}(t)$ has a (fundamental) set of vector functions $\mathbf{x}_j(t) = \boldsymbol{\varphi}_j e^{\omega_j t}$ as solutions (Equation (A4), LHS), where each $\mathbf{x}_j(t)$ combines an eigenvector $\boldsymbol{\varphi}_j$ of \mathbf{A} with an exponential function of the corresponding eigenvalue ω_j of \mathbf{A} (Equation (A4), RHS).

$$\begin{bmatrix} | & | & \dots \\ \mathbf{x}_1(t) & \mathbf{x}_2(t) & \dots \\ | & | & \dots \end{bmatrix} = \begin{bmatrix} | & | & \dots \\ \boldsymbol{\varphi}_1 & \boldsymbol{\varphi}_2 & \dots \\ | & | & \dots \end{bmatrix} \begin{bmatrix} e^{\omega_1 t} & & \\ & e^{\omega_2 t} & \\ & & \ddots \end{bmatrix} \tag{A4}$$

or

$$\mathbf{X}(t) = \boldsymbol{\Phi}\mathbf{Q}(\boldsymbol{\omega}, t)$$

A superposition of these vector functions yields the general solution to the system of differential equations: $\mathbf{x}(t) = \mathbf{X}(t)\mathbf{c}$, where $\mathbf{X}(t)$ is referred to as the fundamental matrix, and \mathbf{c} is a vector of coefficients. Given some initial conditions $\mathbf{x}(0) = \mathbf{x}_0$ and noticing that $\mathbf{X}(0) = \boldsymbol{\Phi}$ (since $\mathbf{Q}(\boldsymbol{\omega}, 0) = \mathbf{I}$), the coefficients \mathbf{c} are determined straightforwardly (from the general solution) $\mathbf{x}_0 = \boldsymbol{\Phi}\mathbf{c} \rightarrow \mathbf{c} = \boldsymbol{\Phi}^\dagger \mathbf{x}_0$. Substituting the expression for \mathbf{c} back into the general solution and using $\mathbf{X}(t) = \boldsymbol{\Phi}\mathbf{Q}(\boldsymbol{\omega}, t)$ (Equation (A4)) finds Equation (A5):

$$\mathbf{x}(t) = \boldsymbol{\Phi}\mathbf{Q}(\boldsymbol{\omega}, t)\boldsymbol{\Phi}^\dagger \mathbf{x}_0 \tag{A5}$$

Now note the similarity between Equations (A2) and (A5): if a frame \mathbf{x}_i (Equation (A2)) = $\mathbf{x}(t_i)$ (Equation (A5)), and recognizing that $\Psi(\omega, t)_{i,j} = Q(\omega, t)_{j,j} = e^{\omega_j t_i}$, then $\boldsymbol{\beta}_j = s\boldsymbol{\varphi}_j$, where the scalar value $s = \boldsymbol{\Phi}_{j,:}^\dagger \mathbf{x}_0$. It proves that the weighted dynamic modes $\boldsymbol{\beta}_j$ resemble scaled eigenvectors $s\boldsymbol{\varphi}_j$ of the linear model \mathbf{A} in the system of differential equations $d\mathbf{x}/dt = \mathbf{A}\mathbf{x}(t)$. Moreover, it shows that Equation (A2) is inherently linked to the differential equation problem Equation (A5) and that given $\mathbf{x}(t_i)$ at sample times t_i , Equation (A3) can hence be used to solve this system of differential equations for \mathbf{A} .

Appendix C. Default Algorithm Settings

Table A2. Default parameter values of the mapping algorithm.

Parameter	Value
N	64 frames
Overlap	16 s (results, Section 4) Variable (on-the-fly, Section 5)
Analytic extension of \mathbf{X}	True
Mode frequencies (= r if \mathbf{X} is analytic)	16
$[T_{min}, T_{max}]$	[3,15] s
Subdomain size per ω^\dagger	$2 \times L_{off}(\omega^\dagger)$ (maximum) $1 \times L_{off}(\omega^\dagger)$ (minimum)
$[\Gamma_{min}, \Gamma_{max}]$	[0.3,1.0]
Rad	75 m
Stationary time (temp. spectral storage)	60 s
α	0.012
$ \mathbf{U} _{max}$	0.75 m/s
$[d_{min}, d_{max}]$	[0.1,50] m
Q_c	0.0005 m ² /s ³
Q_U	0.0005 m ² /s ³

References

1. de Schipper, M.A.; Ludka, B.C.; Raubenheimer, B.; Luijendijk, A.P.; Schlacher, T.A. Beach nourishment has complex implications for the future of sandy shores. *Nat. Rev. Earth Environ.* **2021**, *2*, 70–84. [CrossRef]
2. Gawehn, M.; van Dongeren, A.; de Vries, S.; Swinkels, C.; Hoekstra, R.; Aarninkhof, S.; Friedman, J. The application of a radar-based depth inversion method to monitor near-shore nourishments on an open sandy coast and an ebb-tidal delta. *Coast. Eng.* **2020**, *159*, 103716. [CrossRef]
3. Van Prooijen, B.C.; Tissier, M.F.; De Wit, F.P.; Pearson, S.G.; Brakenhoff, L.B.; Van Maarseveen, M.C.; Van Der Vegt, M.; Mol, J.W.; Kok, F.; Holzhauser, H.; et al. Measurements of hydrodynamics, sediment, morphology and benthos on Ameland ebb-tidal delta and lower shoreface. *Earth Syst. Sci. Data* **2020**, *12*, 2775–2786. [CrossRef]
4. van Dongeren, A.; Ciavola, P.; Martinez, G.; Viavattene, C.; Bogaard, T.; Ferreira, O.; Higgins, R.; McCall, R. Introduction to RISC-KIT: Resilience-increasing strategies for coasts. *Coast. Eng.* **2018**, *134*, 2–9. [CrossRef]
5. Austin, M.J.; Scott, T.M.; Russell, P.E.; Masselink, G. Rip current prediction: Development, validation, and evaluation of an operational tool. *J. Coast. Res.* **2013**, *29*, 283–300. [CrossRef]
6. Davidson, M.; Aarninkhof, S.; Van Koningsveld, M.; Holman, R. Developing coastal video monitoring systems in support of coastal zone management. *J. Coast. Res.* **2006**, *2004*, 49–56.
7. Radermacher, M.; Wengrove, M.; Van Thiel De Vries, J.; Holman, R. Applicability of video-derived bathymetry estimates to nearshore current model predictions. *J. Coast. Res.* **2014**, *70*, 290–295. [CrossRef]
8. Netzband, A.; Adnitt, C. Dredging Management Practices for the Environment: A Structured Selection Approach. *Terra Et Aqua* **2009**, *114*, 3–8.
9. Williams, W. The Determination of Gradients on Enemy-Held Beaches. *Geogr. J.* **1947**, *109*, 76–90. [CrossRef]
10. Holman, R.A.; Stanley, J. The history and technical capabilities of Argus. *Coast. Eng.* **2007**, *54*, 477–491. [CrossRef]
11. Dugan, J.P.; Piotrowski, C.C.; Williams, J.Z. Water depth and surface current retrievals from airborne optical measurements of surface gravity wave dispersion. *J. Geophys. Res. Ocean.* **2001**, *106*, 16903–16915. [CrossRef]
12. Bergsma, E.W.; Almar, R.; Melo de Almeida, L.P.; Sall, M. On the operational use of UAVs for video-derived bathymetry. *Coast. Eng.* **2019**, *152*, 103527. [CrossRef]
13. Brodie, K.L.; Bruder, B.L.; Slocum, R.K.; Spore, N.J. Simultaneous Mapping of Coastal Topography and Bathymetry from a Lightweight Multicamera UAS. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6844–6864. [CrossRef]
14. Holland, K.T.; Lalejini, D.M.; Spansel, S.D.; Holman, R.A. Littoral environmental reconnaissance using tactical imagery from unmanned aircraft systems. *Ocean. Sens. Monit. II* **2010**, *7678*, 767806. [CrossRef]
15. Holman, R.A.; Holland, K.T.; Lalejini, D.M.; Spansel, S.D. Surf zone characterization from Unmanned Aerial Vehicle imagery. *Ocean. Dyn.* **2011**, *61*, 1927–1935. [CrossRef]
16. Holman, R.A.; Brodie, K.L.; Spore, N.J. Surf Zone Characterization Using a Small Quadcopter: Technical Issues and Procedures. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2017–2027. [CrossRef]
17. Bell, P. Shallow water bathymetry derived from an analysis of X-band marine radar images of waves. *Coast. Eng.* **1999**, *37*, 513–527. [CrossRef]
18. Seemann, J.; Ziemer, F.; Senet, C. A method for computing calibrated ocean wave spectra from measurements with a nautical X-band radar. In *Proceedings of the Oceans '97. MTS/IEEE Conference Proceedings, Halifax, NS, Canada, 6–9 October 1997*; IEEE: Piscataway, NJ, USA, 1997; Volume 2, pp. 1148–1154. [CrossRef]
19. Serafino, F.; Lugni, C.; Soldovieri, F. A novel strategy for the surface current determination from marine X-Band radar data. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 231–235. [CrossRef]
20. Young, I.R.; Rosenthal, W.; Ziemer, F. A three-dimensional analysis of marine radar images for the determination of ocean wave directionality and surface currents. *J. Geophys. Res.* **1985**, *90*, 1049–1059. [CrossRef]
21. Ziemer, F. Directional spectra from shipboard navigation radar during LEWEX. In *Directional Ocean Wave Spectra: Measuring, Modeling, Predicting, and Applying*; Beal, R.C., Ed.; The Johns Hopkins University Press: Baltimore, MA, USA, 1991; Volume 1, pp. 125–127.
22. Abileah, R. Mapping near shore bathymetry using wave kinematics in a time series of WorldView-2 satellite images. *Int. Geosci. Remote Sens. Symp. (IGARSS)* **2013**, *2*, 2274–2277. [CrossRef]
23. Almar, R.; Bergsma, E.W.; Maisongrande, P.; de Almeida, L.P.M. Wave-derived coastal bathymetry from satellite video imagery: A showcase with Pleiades persistent mode. *Remote Sens. Environ.* **2019**, *231*, 111263. [CrossRef]
24. Aarninkhof, S.G.J.; Ruessink, B.G.; Roelvink, J.A. Nearshore subtidal bathymetry from time-exposure video images. *J. Geophys. Res. Ocean.* **2005**, *110*, 1–13. [CrossRef]
25. Benschila, R.; Thoumyre, G.; Najar, M.A.; Abessolo, G.; Almar, R.; Bergsma, E.; Hugonnard, G.; Labracherie, L.; Lavie, B.; Ragonneau, T.; et al. A Deep Learning Approach for Estimation of the Nearshore Bathymetry. *J. Coast. Res.* **2020**, *95*, 1011–1015. [CrossRef]
26. Chernyshov, P.; Vrecica, T.; Streßer, M.; Carrasco, R.; Toledo, Y. Rapid wavelet-based bathymetry inversion method for nearshore X-band radars. *Remote Sens. Environ.* **2020**, *240*, 111688. [CrossRef]
27. Collins, A.; Brodie, K.L.; Bak, S.; Hesser, T.; Farthing, M.W.; Gamble, D.W.; Long, J.W. A 2D fully convolutional neural network for nearshore and surfzone bathymetry inversion from synthetic imagery of the surfzone using the wave model Celeris. *CEUR Workshop Proceedings*. **2020**, pp. 1–8. Available online: http://ceur-ws.org/Vol-2587/article_1.pdf (accessed on 1 November 2021).

28. Ghorbanidehno, H.; Lee, J.; Farthing, M.; Hesser, T.; Kitanidis, P.K.; Darve, E.F. Novel data assimilation algorithm for nearshore bathymetry. *J. Atmos. Ocean. Technol.* **2019**, *36*, 699–715. [[CrossRef](#)]
29. van Dongeren, A.; Plant, N.; Cohen, A.; Roelvink, D.; Haller, M.C.; Catalán, P. Beach Wizard: Nearshore bathymetry estimation through assimilation of model computations and remote observations. *Coast. Eng.* **2008**, *55*, 1016–1027. [[CrossRef](#)]
30. Senet, C.M.; Seemann, J.; Flampouris, S.; Ziemer, F. Determination of bathymetric and current maps by the method DiSC based on the analysis of nautical X-band radar image sequences of the sea surface (November 2007). *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2267–2279. [[CrossRef](#)]
31. Honegger, D.A.; Haller, M.C.; Holman, R.A. High-resolution bathymetry estimates via X-band marine radar: 1. beaches. *Coast. Eng.* **2019**, *149*, 39–48. [[CrossRef](#)]
32. Wilson, G.W.; Özkan-Haller, H.T.; Holman, R.A.; Haller, M.C.; Honegger, D.A.; Chickadel, C.C. Surf zone bathymetry and circulation predictions via data assimilation of remote sensing observations. *J. Geophys. Res. Ocean.* **2014**, *119*, 1993–2016. [[CrossRef](#)]
33. Atanassov, V.; Rosenthal, W.; Ziemer, F. Removal of Ambiguity of Two-Dimensional Power Spectra Obtained by Processing Ship Radar Images of Ocean Waves. *J. Geophys. Res.* **1985**, *90*, 1061–1067. [[CrossRef](#)]
34. Borge, J.C.N.; Reichert, K.; Dittmer, J. Use of nautical radar as a wave monitoring instrument. *Coast. Eng.* **1999**, *37*, 331–342. [[CrossRef](#)]
35. Izquierdo, P.; Guedes Soares, C. Analysis of sea waves and wind from X-band radar. *Ocean. Eng.* **2005**, *32*, 1404–1419. [[CrossRef](#)]
36. Wolf, J.; Bell, P.S. Waves at Holderness from X-band radar. *Coast. Eng.* **2001**, *43*, 247–263. [[CrossRef](#)]
37. Irani, G.B.; Gotwols, B.L.; Bjerkaas, A.W. THE 1978 OCEAN WAVE DYNAMICS EXPERIMENT, Optical and in Situ Measurement of the Phase Velocity of Wind Waves. In *Wave Dynamics and Radio Probing of the Ocean Surface*, 1st ed.; Phillips, O., Hasselmann, K., Eds.; Plenum Press: New York, NY, USA; London, UK, 1986; Chapter 10, pp. 165–179.
38. Jähne, B.; Klinke, J.; Geißler, P.; Hering, F. Image Sequence Analysis of Ocean Wind Waves. In *Proceedings of the International Seminar on Imaging in Transport Processes*, Athens, Greece, 25–29 May 1992; pp. 1–12.
39. Stockdon, H.F.; Holman, R.A. Estimation of wave phase speed and nearshore bathymetry from video imagery. *J. Geophys. Res. Ocean.* **2000**, *105*, 22015–22033. [[CrossRef](#)]
40. Hessner, K.; Reichert, K.; Carlos, J.; Borge, N.; Stevens, C.L.; Smith, M.J. High-resolution X-Band radar measurements of currents, bathymetry and sea state in highly inhomogeneous coastal areas. *Ocean. Dyn.* **2014**, *64*, 989–998. [[CrossRef](#)]
41. Senet, C.M.; Seemann, J.; Ziemer, F. The near-surface current velocity determined from image sequences of the sea surface. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 492–505. [[CrossRef](#)]
42. Hessner, K.; Reichert, K.; Rosenthal, W. Mapping of Sea Bottom Topography in Shallow Seas by Using a Nautical Radar. In *Proceedings of the 2nd International Symposium on Operationalization of Remote Sensing*, Enschede, The Netherlands, 16–20 August 1999; pp. 1–5.
43. Holman, R.; Plant, N.; Holland, T. CBathy: A robust algorithm for estimating nearshore bathymetry. *J. Geophys. Res. Ocean.* **2013**, *118*, 2595–2609. [[CrossRef](#)]
44. Zuckerman, S.; Anderson, S. Bathymetry and water-level estimation using x-band radar at a tidal inlet. *J. Coast. Res.* **2018**, *34*, 1227–1235. [[CrossRef](#)]
45. Heathershaw, A.D.; Blackley, M.W.; Hardcastle, P.J. Wave direction estimates in coastal waters using radar. *Coast. Eng.* **1979**, *3*, 249–267. [[CrossRef](#)]
46. Mattie, M.G.; Lee Harris, D. The Use of Imaging Radar in Studying Ocean Waves. *Coast. Eng.* **1978**, *1*, 174–189.
47. Bell, P. Mapping Shallow Water Coastal Areas Using a Standard Marine X-Band Radar. In *Hydro8*; International Federation of Hydrographic Societies: Liverpool, UK, 2008; pp. 1–9.
48. Honegger, D.A.; Haller, M.C.; Holman, R.A. High-resolution bathymetry estimates via X-band marine radar: 2. Effects of currents at tidal inlets. *Coast. Eng.* **2020**, *156*, 103626. [[CrossRef](#)]
49. Ludeno, G.; Reale, F.; Dentale, F.; Carratelli, E.P.; Natale, A.; Soldovieri, F.; Serafino, F. An X-band radar system for bathymetry and wave field analysis in a harbour area. *Sensors* **2015**, *15*, 1691–1707. [[CrossRef](#)] [[PubMed](#)]
50. Campana, J.; Terrill, E.J.; de Paolo, T. The development of an inversion technique to extract vertical current profiles from X-band radar observations. *J. Atmos. Ocean. Technol.* **2016**, *33*, 2015–2028. [[CrossRef](#)]
51. Smeltzer, B.K.; Æsøy, E.; Ådnøy, A.; Ellingsen, S. An Improved Method for Determining Near-Surface Currents From Wave Dispersion Measurements. *J. Geophys. Res. Ocean.* **2019**, *124*, 8832–8851. [[CrossRef](#)]
52. Hessner, K.; Wallbridge, S.; Dolphin, T. Validation of areal wave and current measurements based on X-band radar. In *Proceedings of the 2015 IEEE/OES 11th Current, Waves and Turbulence Measurement, CWTM 2015*, St. Petersburg, FL, USA, 2–6 March 2015; pp. 1–10. [[CrossRef](#)]
53. Simarro, G.; Calvete, D.; Luque, P.; Orfila, A.; Ribas, F. UBathy: A new approach for bathymetric inversion from video imagery. *Remote Sens.* **2019**, *11*, 2722. [[CrossRef](#)]
54. Brunton, S.L.; Proctor, J.L.; Tu, J.H.; Kutz, J.N. Compressed sensing and dynamic mode decomposition. *J. Comput. Dyn.* **2015**, *2*, 165–191. [[CrossRef](#)]
55. Tu, J.H.; Rowley, C.W.; Luchtenburg, D.M.; Brunton, S.L.; Kutz, J.N. On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.* **2014**, *1*, 391–421. [[CrossRef](#)]
56. Jovanović, M.R.; Schmid, P.J.; Nichols, J.W. Sparsity-promoting dynamic mode decomposition. *Phys. Fluids* **2014**, *26*. [[CrossRef](#)]

57. Proctor, J.L.; Brunton, S.L.; Kutz, J.N. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* **2016**, *15*, 142–161. [[CrossRef](#)]
58. Kutz, J.N.; Fu, X.; Brunton, S.L. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **2016**, *15*, 713–735. [[CrossRef](#)]
59. Klus, S.; Gelß, P.; Peitz, S.; Schütte, C. Tensor-based dynamic mode decomposition. *Nonlinearity* **2018**, *31*, 3359–3380. [[CrossRef](#)]
60. Askham, T.; Kutz, J.N. Variable projection methods for an optimized dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **2018**, *17*, 380–416. [[CrossRef](#)]
61. Barocio, E.; Pal, B.C.; Thornhill, N.F.; Messina, A.R. A Dynamic Mode Decomposition Framework for Global Power System Oscillation Analysis. *IEEE Trans. Power Syst.* **2015**, *30*, 2902–2912. [[CrossRef](#)]
62. Mohan, N.; Soman, K.P.; Sachin Kumar, S. A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model. *Appl. Energy* **2018**, *232*, 229–244. [[CrossRef](#)]
63. Brunton, B.W.; Johnson, L.A.; Ojemann, J.G.; Kutz, J.N. Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *J. Neurosci. Methods* **2016**, *258*, 1–15. [[CrossRef](#)]
64. Wang, R.q.; Herdman, L.M.; Erikson, L.; Barnard, P.; Hummel, M.; Stacey, M.T. Interactions of Estuarine Shoreline Infrastructure With Multiscale Sea Level Variability. *J. Geophys. Res. Ocean.* **2017**, *122*, 9962–9979. [[CrossRef](#)]
65. Taylor, Z.J.; Gurka, R.; Kopp, G.A.; Liberzon, A. Long-duration time-resolved PIV to study unsteady aerodynamics. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 3262–3269. [[CrossRef](#)]
66. Holthuijsen, L.H. *Waves in Oceanic and Coastal Waters*; Cambridge University Press: Cambridge, UK, 2007.
67. Barron, J.T. A general and adaptive robust loss function. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4326–4334. [[CrossRef](#)]
68. Madsen, K.; Nielsen, H.B.; Tingleff, O. *Methods for Non-Linear Least Squares Problems*, 2nd ed.; Technical University of Denmark: Copenhagen, Denmark, 2004; pp. 1–58.
69. Black, M.J.; Rangarajan, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision* **1996**, *19*, 57–91. [[CrossRef](#)]
70. Rutten, J.; De Jong, S.M.; Ruessink, G. Accuracy of Nearshore Bathymetry Inverted from X-Band Radar and Optical Video Data. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 1106–1116. [[CrossRef](#)]
71. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [[CrossRef](#)]
72. Gavish, M.; Donoho, D.L. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Inf. Theory* **2014**, *60*, 5040–5053. [[CrossRef](#)]
73. Mendible, A.; Koch, J.; Lange, H.; Brunton, S.L.; Kutz, J.N. Data-driven modeling of rotating detonation waves. *Phys. Rev. Fluids* **2021**, *6*, 1–20. [[CrossRef](#)]
74. Askham, T.; Zheng, P.; Aravkin, A.; Kutz, J.N. Robust and scalable methods for the dynamic mode decomposition. *arXiv* **2017**, arXiv:1712.01883.
75. Elgar, S.; Herbers, T.H.C.; Guza, R.T. Reflection of Ocean Surface Gravity Waves from a Natural Beach. *J. Phys. Oceanogr.* **1994**, *24*, 1503–1511. [[CrossRef](#)]
76. Brodie, K.L.; Raubenheimer, B.; Elgar, S.; Slocum, R.K.; McNinch, J.E. Lidar and pressure measurements of inner-surfzone waves and setup. *J. Atmos. Ocean. Technol.* **2015**, *32*, 1945–1959. [[CrossRef](#)]
77. Huntley, D.A.; Simmonds, D.; Tataavarti, R. Use of Collocated Sensors to Measure Coastal Wave Reflection. *J. Waterw. Port Coastal Ocean. Eng.* **1999**, *125*, 46–52. [[CrossRef](#)]
78. Battjes, J.A. Surf similarity. In Proceedings of the 14th International Conference on Coastal Engineering, Copenhagen, Denmark, 24–28 June 1974; Volume 1, pp. 466–480.
79. Raubenheimer, B.; Guza, R.T. Observations and predictions of run-up. *J. Geophys. Res. Ocean.* **1996**, *101*, 25575–25587. [[CrossRef](#)]
80. Miles, J.R.; Russell, P.E.; Huntley, D.A. Sediment transport and wave reflection near a seawall. *Proc. Coast. Eng. Conf.* **1997**, *3*, 2612–2624. [[CrossRef](#)]
81. Goda, Y.; Suzuki, Y. Estimation of incident and reflected waves in random wave experiments. In *Coastal Engineering 1976*; ASCE: Honolulu, HI, USA, 1976; pp. 828–845.
82. Gawehn, M.; de Vries, S.; Aarninkhof, S. Depth and Surface Current Inversion on the Fly: A New Video Based Approach Using the Dynamic Mode Decomposition. In Proceedings of the Coastal Sediments 2019—Proceedings of the 9th International Conference, St. Petersburg, FL, USA, 27–31 May 2019; pp. 2511–2520. [[CrossRef](#)]
83. Trizna, D. Errors in bathymetric retrievals using linear dispersion in 3-D FFT analysis of marine radar ocean wave imagery. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 2465–2469. [[CrossRef](#)]
84. Fienup, J.R. Invariant error metrics for image reconstruction. *Appl. Opt.* **1997**, *36*, 8352. [[CrossRef](#)]
85. Bosboom, J.; Stive, M.J.F. *Coastal Dynamics*; Delft University of Technology: Delft, The Netherlands, 2021; Volume 1269, p. 595. [[CrossRef](#)]
86. Bell, P.S.; Osler, J.C. Mapping bathymetry using X-band marine radar data recorded from a moving vessel. *Ocean. Dyn.* **2011**, *61*, 2141–2156. [[CrossRef](#)]
87. Kraft, D. *A Software Package for Sequential Quadratic Programming*; Technical Report; DLR German Aerospace Center—Institute for Flight Mechanics: Köln, Germany, 1988.

88. Bergsma, E.W.; Conley, D.C.; Davidson, M.A.; O'Hare, T.J. Video-based nearshore bathymetry estimation in macro-tidal environments. *Mar. Geol.* **2016**, *374*, 31–41. [[CrossRef](#)]
89. Aarnink, J.L. Bathymetry Mapping Using Drone Imagery. Master's Thesis, Delft University of Technology, Delft, The Netherlands, 2017.
90. Vos, K. Remote Sensing of the Nearshore Zone Using a Rotary-Wing UAV. Master's Thesis, University of New South Wales, Kensington, Australia, 2017.
91. Bergsma, E.W.J. Application of an Improved Video-Based Depth Inversion Technique to a Macrotidal Sandy Beach. Ph.D. Thesis, University of Plymouth, Plymouth, UK, 2017.
92. Bell, P. Determination of bathymetry using marine radar images of waves. In Proceedings of the 4th International Symposium on Ocean Wave Measurement and Analysis, San Francisco, CA, USA, 2–6 September 2001; pp. 251–257. [[CrossRef](#)]
93. Reniers, A.J.; Thornton, E.B.; Stanton, T.P.; Roelvink, J.A. Vertical flow structure during Sandy Duck: Observations and modeling. *Coast. Eng.* **2004**, *51*, 237–260. [[CrossRef](#)]
94. Sonu, C.J. Field observation of nearshore circulation and meandering currents. *J. Geophys. Res.* **1972**, *77*, 3232–3247. [[CrossRef](#)]
95. Short, A.D. Australian rip systems—Friend or foe? *J. Coast. Res.* **2007**, *2007*, 7–11.
96. Anderson, D.; Spicer Bak, A.; Brodie, K.L.; Cohn, N.; Holman, R.A.; Stanley, J. Quantifying optically derived two-dimensional wave-averaged currents in the surf zone. *Remote Sens.* **2021**, *13*, 690. [[CrossRef](#)]
97. Haller, M.C.; Honegger, D.; Catalan, P.A. Rip Current Observations via Marine Radar. *J. Waterw. Port Coastal Ocean. Eng.* **2014**, *140*, 115–124. [[CrossRef](#)]
98. Mortlock, T.R.; Goodwin, I.D.; McAneney, J.K.; Roche, K. The June 2016 Australian East Coast Low: Importance of wave direction for coastal erosion assessment. *Water* **2017**, *9*, 121. [[CrossRef](#)]
99. Plant, N.G.; Holland, K.T.; Haller, M.C. Ocean wavenumber estimation from wave-resolving time series imagery. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2644–2658. [[CrossRef](#)]
100. Boyce, W.E.; DiPrima, R.C. *Elementary Differential Equations and Boundary Value Problems*, 9th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.