https://doi.org/10.1007/s41651-021-00085-8

An Image-Based Framework for Ocean Feature Detection and Analysis

Divya Banesh[1,2] · Mark R. Petersen[2] · James Ahrens[2] · Terece L. Turton[2] · Francesca Samsel[2] · Joseph Schoonover[3] · Bernd Hamann[1]

Accepted: 28 June 2021

Abstract

Today's supercomputing capabilities allow ocean scientists to generate simulation data at increasingly higher spatial and temporal resolutions. However, I/O bandwidth and data storage costs limit the amount of data saved to disk. In situ methods are one solution to generate reduced data extracts, with the potential to reduce disk storage requirement even for high spatial and temporal resolutions, a major advantage to storing raw output. *Image proxies* have become an efficient and accepted in situ reduced data extract. These extracts require innovative automated techniques to identify and analyze features. We present a framework of computer vision and image processing techniques to detect and analyze important features from in situ image proxies of large ocean simulations. We constrain the analysis framework in support of techniques that emulate ocean-specific tasks as accurately as possible. The framework maximizes feature analysis capabilities while minimizing computational requirements. We demonstrate its use for image proxies extracted from the ocean component of Model for Prediction Across Scales (MPAS) simulations to analyze ocean-specific features such as eddies and western boundary currents. The results obtained for specific data sets are compared to those of traditional methods, documenting the efficacy and advantages of our framework.

Keywords Computer vision · In situ analysis · MPAS · Feature detection

Introduction

Science is experiencing the era of large data, presenting numerous benefits and challenges. In the ocean science community, more advanced computing capabilities allow scientists to conduct simulations at ever higher resolutions and for increasingly longer time periods. For example, high-resolution ocean simulations can now include 3.7 million cells in two-dimensional layers, with 80 vertical layers stacked on top of each other (Petersen et al. 2019), consider thirty variables at 5-day to monthly intervals, and simulate behavior for decades or even centuries. Data output from atmosphere, sea ice, and land components in earth system models are of similar scale (Caldwell et al. 2019).

More powerful computing capabilities also lead to *challenges*. Limited I/O bandwidth makes visualization and analysis of large simulations difficult and often inefficient. Data storage costs limit the size and number of time steps that can be stored to only a small percentage of the simulation (Bauer et al. 2016). Scientists cannot mentally envision the entirety of such amounts of data and must

✉ Divya Banesh
dbanesh@ucdavis.edu

Mark R. Petersen
mpetersen@lanl.gov

James Ahrens
ahrens@lanl.gov

Terece L. Turton
tlturton@lanl.gov

Francesca Samsel
samsel@lanl.gov

Joseph Schoonover
schoonover.numerics@gmail.com

Bernd Hamann
hamann@cs.ucdavis.edu

[1] University of California, Davis, Davis, CA, USA

[2] Los Alamos National Laboratory, Los Alamos, NM, USA

[3] NOAA, Boulder, CO, USA

Published online: 02 August 2021

evaluate their simulation outputs by visualizing averages or subsets of a full data set (Golaz et al. 2019). Researchers can employ various data reduction techniques, such as compression, data sampling, in situ methods and statistical tools, to reduce the size of simulation data to a manageable size (Li et al. 2018). Compression techniques can drastically reduce the data saved to disk but are generally not feature-driven, which can lead to undesirable data and thus information loss. Data sampling methods can identify regions of interest and intelligently preserve information in those regions while aggressively discarding regions not containing relevant information (Dutta et al. 2019).

Image proxies have emerged recently as an additional means to preserve important features while reducing data size. Tikhonova et al. (2010b) introduced proxy images for reduction of volumetric data. The method captures depth information for various isovalues, using a volume distortion camera model, and saves image data in a modified format. Similarly, techniques such as Cinema (Ahrens et al. 2015; Kageyama and Yamada 2014) project simulation data to a viewing plane, effectively compressing the simulation information via a high-resolution image. Wang et al. (2018, 2019) introduced image proxies combined with statistical distributions to support high-accuracy re-rendering, apply transfer functions and extract iso-surfaces from volumetric data. The effectiveness of each one of these methods to address concerns with large data analysis are discussed in the associated papers. However, with the success of such techniques also arises the need for tools that can extract features of interest from image proxies, rather than requiring that a scientist must re-consider and perform a computationally more expensive simulation for an in-depth post hoc analysis. These tools must provide methods that make it possible to precisely and accurately identify, extract and analyze important, scientifically correct features in large simulation data sets.

We address these needs via a *framework of image processing and computer vision techniques* that allows one to extract scientific features for analysis using a compressed proxy format to maximize feature analysis while minimizing computational requirements. To ensure an accurate representation of simulation data, the design of our framework was driven by two requirements: (1) An effective analysis and visualization of image proxies must replicate techniques and identify features similar to those used and identified by a scientist when working with the original simulation data. Such a design encourages scientists to regard the proxies as an extension of the original data rather than completely different entities. (2) Considering the use of image proxies for simulation studies, the framework must leverage traditional image processing and computer vision to provide additional desired capabilities to a scientist. A framework satisfying this requirements allows a scientist to take full advantage of 2D representations.

The framework presented here satisfies these requirements. We present a visualization framework that uses computer vision feature analysis tools to extract and track scientifically relevant features. Four main elements are included in the feature analysis toolkit: (1) image filters to reduce the effects of high-frequency noise, (2) feature detectors to identify science-specific features, (3) feature matching methods to gain insight from one image to another, and (4) feature tracking methods to understand feature evolution. The results generated by specific methods selected from these capabilities can be viewed interactively via the framework's real-time data exploration interface, which supports statistical methods and graphing tools for robust and detailed analysis.

While other ocean simulation analysis systems provide similar capabilities to the framework presented, we find that none provide the same *integrated combination* of features. Paraview (Ahrens et al. 2005), for example, provides an excellent UI with numerous feature detection and visualization methods, but little support for feature matching and tracking. ImageJ (Schneider et al. 2012) supports many image processing needs, but none of the more advanced computer vision algorithms included here. In addition, the statistical techniques in both these frameworks do not include those presented. Methods we include are tailored to and driven by the needs of ocean scientists. The framework presented also provides scientists a more direct and interactive control over how algorithms are applied. For example, our UI allows scientists to threshold the results of analysis algorithms to emphasize scientific meaning. Although this is theoretically possible with the other systems, there is no native support and must be provided through a scripting interface. In addition, the small data footprint of image proxies makes the framework viable. A scientist does not have to wait minutes or hours for data to be re-rendered as a result of changes to an algorithm or parameter value; updates are generated in seconds. The framework directly enables a highly exploratory scientific analysis process.

We illustrate the use of our framework in two case studies: (1) the detection and analysis of the Northern Boundary of the Gulf Stream in an ocean simulation and (2) the identification and tracking of ocean eddies near the Agulhas Current in an ocean simulation. We evaluate the results of each study through a comparison with traditional analysis methods commonly used by ocean scientists. Our comparisons demonstrate the utility of our framework as an alternative to traditional methods.

In summary, our contributions to ocean science simulation data analysis are:

1. An analysis framework that allows scientists to emulate techniques used in traditional domain-specific tasks and extract features typical to their scientific domain.
2. The use of computer vision algorithms to provide additional capabilities for the use of images as a means for scientific analysis.
3. The evaluation of domain-specific features and a comparison to traditional analysis methods to validate the presented visualization framework.

Related Work

Frameworks for Image Analysis Once data is in image format, numerous frameworks are available for data exploration. ImageJ (Schneider et al. 2012) is a popular toolkit with various image processing capabilities; however, it does not 4 inherently provide matching and tracking algorithms to understand a feature's behavior over time. Trackmate (Tinevez et al. 2017), a plugin extension to ImageJ, expands on the latter's capabilities through feature detection and tracking. Unfortunately, Trackmate only detects one type of feature, "blobs," as defined by four related 4 algorithms based on the Difference of Gaussians and the Laplacian of Gaussians. However, a gradient-based blob detection algorithm can not be applicable or mathematically rigorous for every feature of interest. Other techniques will be needed to analyze linear features, features comprised of multiple components, groups of features or even blobs better described by closed isovalues rather than a gradient-based method. A second framework, the VAICo framework (Schmidt et al. 2013) identifies regions of differences between pairs of images using mean squared error (MSE) to identify discrepancies. This technique assumes that a majority of the image remains the same, with only smaller regions undergoing change. This is not an assumption our scientists can guarantee due to the amount of noise and turbulence generally present in their data. Such a comparison technique could lead to the identification of many false features and therefore is too simple a method for our needs; more complex feature descriptors are generally required. In addition, the inclusion of more complex image quality metrics such as the structural similarity index measure (SSIM) (Wang et al. 2004) or peak signal-to-noise ratio (PSNR) would result in a more robust framework. A final image analysis framework, PorosityAnalyzer (Weissenböck et al. 2016), compares results from various algorithms, e.g., different blob detection algorithms, to determine which is most useful for a particular application. We expect this sort of comparison to be useful in future work once scientists are more comfortable with image proxy analysis techniques.

Feature Detection and Tracking Feature detection and tracking (Aigner et al. 2011), even when limited to applications in visualization of simulation data, is a broad topic. Tracking methods in visualization cover optical flow, fluid flow studies (Post et al. 2003), feature tracking in the context of merge trees (Widanagamaachchi et al. 2012) and Morse-Smale complexes (Lukasczyk et al. 2017), and others. Early studies by Samtaney et al. (1994) discussed a method using contours for object detection and tracking in images. Reinders et al. (2001) explored more deeply the topic of feature *event* detection to identify time points of births, deaths, splits and merges of features. Liu et al. (2013) and Ozer et al. (2014) developed activity mapping tools to parse data sets and extract specific time steps of interest, based on Silver and Wang (1997). Muelder and Ma (2009) utilized a prediction-correction technique to identify regions containing features of interest. Our framework focuses on the contributions detection and tracking algorithms can make to the field of visualization rather than the reverse.

Data Workflow

To ensure that data is accurately studied, a visualization workflow must have several attributes:

1. The framework must be flexible enough to process data from *multiple data sources*.
2. The framework must ensure that the *accuracy of the inputs* are maintained.
3. The framework must allow for *task specific functionality*.

We address the first two attributes in this section. The final attribute is discussed in Section "Analysis Framework" with examples presented in Section "Case Study Results and Discussion".

The framework presented in Section "Analysis Framework" accommodates data from multiple sources through the use of Cinema (Ahrens et al. 2015; Turton et al. 2020). Cinema is a database of artifacts, such as images or image proxies, with an accompanying CSV file that lists data parameters for each artifact in the database. Processes to generate Cinema databases are varied and can be dependent on the scientists' workflow. For example, we can generate Cinema databases of simulation results through various applications and platforms, including Paraview (Ahrens et al. 2005) and VisIt (Childs and et al. 2012). This can be performed either in post-processing if the simulation data has already been computed, or in situ to the simulation computation on architectures such as ORNL's Summit and LLNL's Sierra (Turton et al. 2020).

Cinema was chosen over other image proxy methods, such as Tikhonova et al. (2010b) and Wang et al. (2018, 2019), for multiple reasons, including, for example: (1) Cinema is an open-source tool and professionally maintained; (2) Cinema is readily available for various platforms, applications and architectures; and (3) Cinema is based on images, thus providing a highly effective and scientifically acceptable way to store and explore complex simulation data using a simple image format, see (DSS Group at LANL 2021) for details.

Simulation Data

In this paper, image proxies of simulation results are generated using Paraview's Cinema Export functionality: Virtual cameras are positioned at theta- and phi-implied positions, using spherical coordinates, encircling the simulation domain, Fig. 1(a). The simulation is projected onto a plane at the camera's location. The data recorded at each pixel of the plane can be simply rendered with colormap and lighting information, or the floating-point value of the simulation saved in the 32 bits provided by the RGBA memory, Fig. 1(b). This latter representation, referred to as a *floating-point image*, maintains the *accuracy* of the simulation values in the image proxies.

The 32-bit floating-point representation is critical for processing simulation data in our analysis toolkit. It allows users to apply feature evaluation algorithms directly to the values of the simulated data, creating a robust system that is not susceptible to distortion due to colormaps or from lighting effects. All feature evaluation techniques are computed directly on these data values and results are colormapped only at the end of the processing pipeline for visual presentation. This ensures that detected, matched, and tracked features are a close representation of events in the simulation. However, the user can opt to apply the same techniques and evaluation to a simply rendered image if desired.

The application of the visualization framework to simulation data is studied here through the use of MPAS-Ocean, a component of the Model for Prediction Across Scales (MPAS) (Ringler et al. 2013; Petersen et al. 2015). MPAS-Ocean is a multi-resolution model of the Earth's oceans and part of the Energy Exascale Earth System Model (E3SM), developed by the US Department of Energy (Golaz et al. 2019). The MPAS-Ocean simulation studied here is tessellated using Voronoi grids at 15-km resolution and contains 173 time-dependent outputs, each 5 days apart. The Cinema database of this data, generated through Paraview, produces a floating-point image at 15 evenly spaced locations in each angle phi and theta at each of the 173 temporal outputs. The simulation is also over-sampled to ensure that each data element is represented by several image pixels, ensuring high-quality input for analysis. The Paraview export functionality tags each image derived from the simulated data set with the corresponding phi, theta, time, and corresponding simulation parametric values. It also generates a CSV file to catalog the saved artifacts. The open-source MPAS-Ocean ocean model that produced the original data used in this study is available at (COSIM Group at LANL & NCAR 2019). The raw model output for high-resolution global simulations is available for download at https://e3sm.org/data/get-e3sm-data/released-e3sm-data/v1-025-data-hiresmip/ (E3SM Project 2018).

Analysis Framework

The visualization framework analyzes images ordered into a Cinema database. The framework allows users to extract and evaluate features through various techniques. The results of these evaluations can be visualized by the provided interface and written back to the Cinema database for further study or for collaborative efforts. The specific components of the visualization framework are shown in Fig. 2.

The visualization framework has two main components: the feature analysis toolkit and the user interface (UI). The feature analysis toolkit operates on a database of images (in RGB or floating-point format) and contains four types of visualization elements: image filters, feature detectors, feature matching methods, and techniques for feature tracking. Each is an optional piece of the user's pipeline dependent on the final goal. Users select methods to form a customized pipeline suitable for their analysis

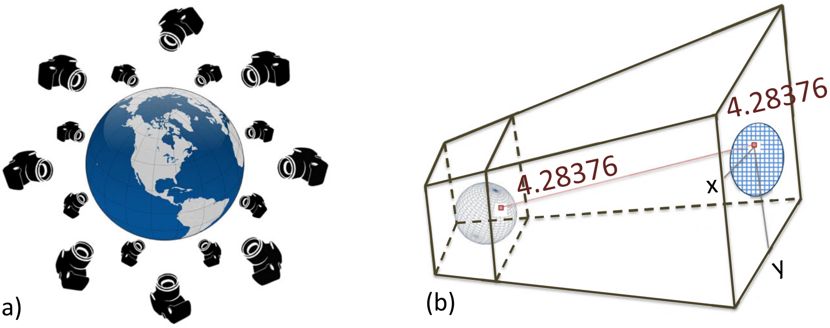


Fig. 1 Cinema floating-point image databases. (a) Images from computer simulation, taken at evenly spaced theta and phi angle locations on a bounding sphere. (b) Projection of simulated data onto an image plane. For every image pixel, the corresponding data value is saved to generate a floating-point image

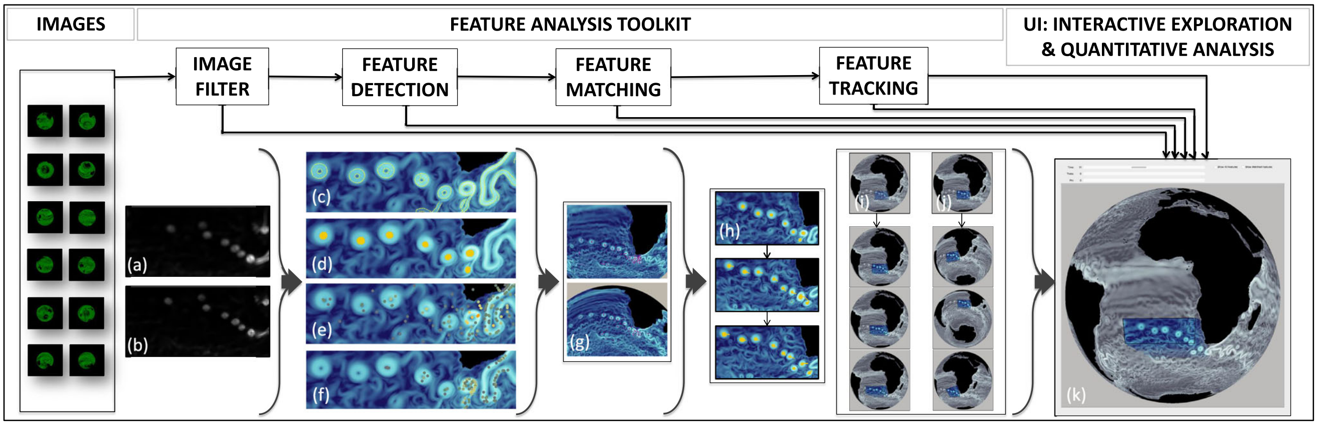


Fig. 2 The five elements of the image-based visualization framework: image filters, feature detection, feature matching, feature tracking, and interactive exploration with quantitative analysis. Using a database of images or image proxies, the user can apply a linear (a) or non-linear (b) filter, select a feature detection algorithm (c-f), select a feature matching algorithm (g), track the features temporally (h),(i) or spatially (j), and view the results in the interactive user interface with any extracted quantitative information (k)

from the available options in each element category. Users can then interact with the results of their selected algorithms within the associated UI, where parametric values can be selected to narrow features to those of scientific interest. Users can also select regions of interest within the UI to capture a subset of features for statistical study, numerical feedback and other quantitative analyses. The feature analysis toolkit uses components from both the Insight Toolkit (Johnson et al. 2013), to support floating-point image representations, and OpenCV (Bradski 2000), for computer vision algorithms. This format allows the visualization framework to be flexible, as scientists can easily integrate new algorithms from these external toolkits into the framework should the scientist require such capabilities.

By separating the image readers and visualization interface from the feature analysis toolkit, it is possible to add one's own custom-built feature analysis pipelines into the framework. Machine-learning or deep-learning algorithms could also be incorporated.

To facilitate a real-time interface, we include a data-tracking component in the framework to record the chronological history of features when evaluating temporal data. Once a feature detection algorithm has been selected and given a set of parametric values for the detection and image filters, the framework records the features detected at a time t. Therefore, if we require information from that time step in the future, the algorithm gathers the information from the saved data rather than recomputing the features. Additionally, any jump forward in time from an initial time t to $t + n$, where $n \geq 1$ results in the computation of all features for time t to $t + n$ at once. Both these techniques ensure continuity in the results and support for robust matching and tracking of features. Furthermore, it is often the case that scientists scroll forwards and backward through smaller temporal sections of their data to evaluate feature changes. Our data-tracking component makes such an evaluation possible and immediate by improving the framework's response time. However, changing the phi-theta viewing angle or altering the algorithmic parameters initiates a reset to the data structure. In such a situation, the framework retroactively computes the results for the new combination of parameters from time 0 to n. Depending on the size of the data and parametric combination, this might result in a small initial delay, but the subsequent analysis is again instantaneous.

We present the various elements of the feature analysis toolkit and user interface in the remainder of Section "Analysis Framework" through their application to MPAS-Ocean simulation results of sea surface kinetic energy. The focus is on mesoscale ocean eddies formed by the Agulhas Retroflection, starting at the coast of Southern Africa and moving west through the Atlantic Ocean, Fig. 3. Eddies are vortices in the ocean with high rotational and translational energy and, therefore, well-defined in the sea surface kinetic energy parameter (Wu et al. 2007).

Image Filters

The first element the scientist must consider is the type of image filter to use. Image filters can significantly enhance visual understanding by, for example, reducing extraneous noise. The framework supports four smoothing filters: *box or mean, Gaussian, median and bilateral*. Box and Gaussian are linear filters, whereas median and bilateral filters are anisotropic. Anisotropic filters are known to detect and preserve edges in image data better than linear filters (Forsyth and Ponce 2011). However, the scientific community more commonly uses box and Gaussian functions. Users can interactively define a window size at zero (filter has no impact)

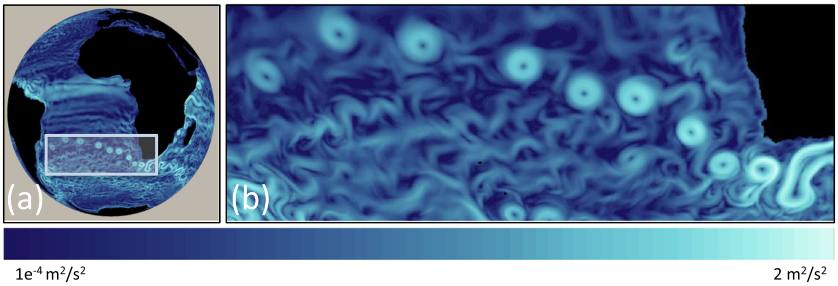


Fig. 3 The Agulhas rings in an MPAS-ocean simulation. Colormapped image of the MPAS-Ocean Simulation (sea surface kinetic energy). The region of interest is highlighted in (a) and expanded in (b). Circular mesoscale eddies, major currents and ocean turbulence are plainly visible

or higher. Filters are especially useful when combined with other methods of the feature analysis toolkit. Figure 4, for example, shows the effects of a Gaussian blur filter on an edge detection algorithm. Adjusting the blur parameter allows the user to minimize scientifically irrelevant features. Edges corresponding to eddies and high kinetic energy currents remain.

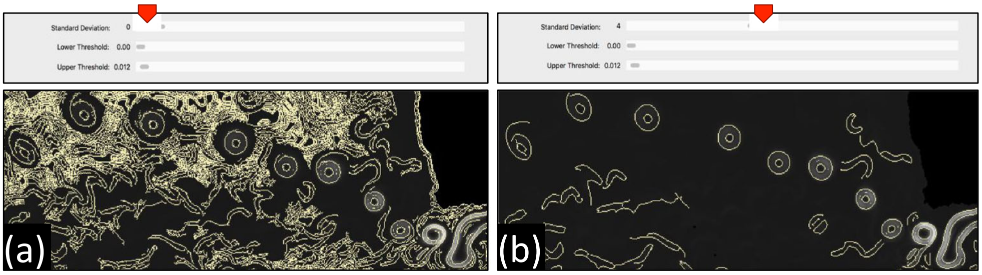
Feature Detection

An almost infinite number of feature detectors exist for evaluation of regions of interest in 2D images. However, the feature detectors we chose as part of this visualization framework were selected due to their similarity to traditional evaluation methods from scientists' workflows. Additionally, we require that features descriptors in the framework allow features to be defined with a high degree of invariance for successful matching and tracking. The features detector techniques included in this framework are as follows: Canny edge detection, image segmentation by a user-defined threshold, Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Features from Accelerated Segment Test (FAST) local feature detectors.

The framework uses *Canny edge detection* to identify regions of rapid change in adjacent pixel values. This edge detection algorithm is based on the Sobel operator, with a low error rate, utilizing non-maximum suppression (Canny 1986). Non-maximum suppression removes an edge's extraneous pixels so the edges are represented as thin and precise as possible. Figure 5(a) shows the results of Canny edge detection to highlight strong discontinuities in the Agulhas region that correspond to eddies and currents.

Contour segments in the framework are defined as closed regions delineated by a poly-line that enclose either a topological super-level or sublevel set. Contours of interest are identified by a user-defined threshold value, setting all pixel values below the threshold to zero and all pixel values above the threshold to one, or vice versa depending on the data and features to identify. A *border following* (Suzuki et al. 1985) algorithm defines a closed contour that, when filled in, determines a 2D region bounded by a closed polyline. Figure 5(b) uses this segmentation technique to identify eddies; non-closed features such as currents are generally missed.

In many circumstances, scientists can not describe features of interest in simple terms, such as by an isovalue. The feature definition might be more abstract, where scientists are looking for regions with certain geometric shapes or clusters of features. For such circumstances, we include abstract feature detectors such as *SIFT* (Lowe 2004), *SURF* (Bay et al. 2006) and *FAST* (Rosten and Drummond 2006) to the framework. The features detected by these algorithms are regions of pixels with high contrast to the surrounding image. Such features are advantageous for tasks that require identification of abstract features with high degrees of invariance that can be matched or tracked to other regions of the simulation, either spatially or temporally. FAST is a corner detection algorithm, while SIFT and SURF are more generalized methods. SIFT is invariant to scaling and rotation, but it does not respond well to noise and is computationally less efficient. SURF is more efficient but sensitive to viewpoint and illumination settings. FAST does not support changes of rotation or scale in its detection (Forsyth and Ponce 2011). We can also view these features as seed points in a flow model, where only strong


Standard Deviation: 0
Lower Threshold: 0.00
Upper Threshold: 0.012

Standard Deviation: 4
Lower Threshold: 0.00
Upper Threshold: 0.012

Fig. 4 Impact of the Guassian filter on feature selection. Changing the window size of a Gaussian filter from 0 (a) to 4 (b) significantly affects a visualization based on edge detection. High-frequency edges are filtered, and lower-frequency edges corresponding to eddies and currents remain

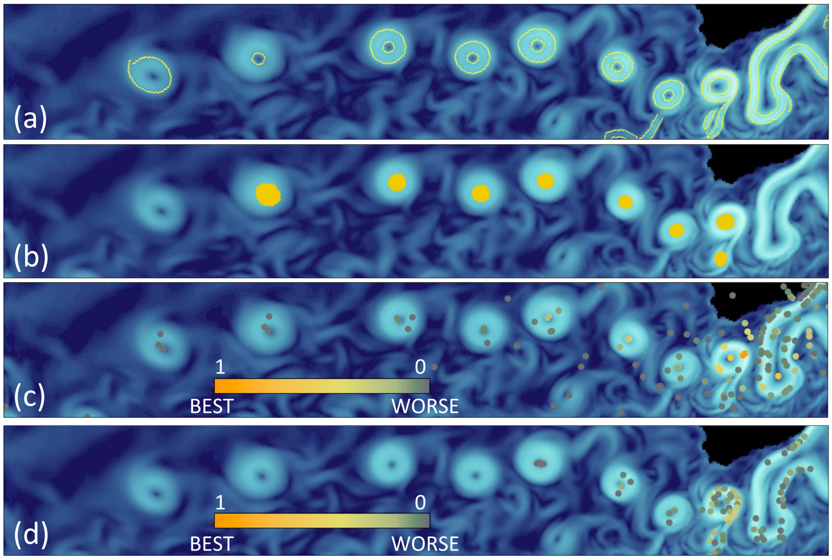


Fig. 5 Four local feature detection methods. Four visualization techniques highlighting attributes of currents and eddies in the Agulhas region (rotated by -17° to enhance comparison). Canny edge detection (a) identifies strong discontinuities, such as high-velocity currents and eddies in the lower-paced ocean. Image segmentation (b) detects eddies due to their shape and uniform kinetic energy. The feature points detected with SURF (c) and FAST (d) are colored with an orange–green colormap. The brighter a point, the stronger the associated feature. The relatively higher contrast of the eddies and currents allows SURF (c) to detect features representing most eddies in the region. FAST (d) only detects features corresponding to higher-velocity features due to their strongly-defined corners

features of interest are seeded and tracked to understand movement over time. In Fig. 5(c) and (d), stronger features tag regions of higher contrast where matching and tracking algorithms are most likely to be successful.

Feature Matching

Gaining more insight from detected features requires effective matching of features from one image to another. Feature matching requires (1) feature descriptions to define feature attributes of interest and (2) a matching algorithm. Within the framework, we define feature attributes with a vector. For each reference feature, the matching algorithm finds the best match by comparing its vector representation to vectors of features in a paired image. Local feature detectors such as SIFT or SURF have methods to define feature descriptions built into the detection algorithm. Other feature attributes a matching algorithm might use includes a feature location or center, feature pixel intensities, feature area, and derived quantities such as Hu moments (1962).

A *brute-force* matching algorithm compares the description for each reference feature with each feature description in a paired image. Using a pre-defined vector-distance metric, we find a match when two feature vectors minimally deviate, when considering all possible pairs. This is an $O(n^2)$ operation, where n is the number of features in an image. When analyzing large feature sets, one should consider a specialized data structure to reduce complexity and ensure real-time system behavior. One such solution included in the framework is *FLANN (Fast Library for Approximate Nearest Neighbors)* (Muja and Lowe 2009): a library of nearest-neighbor algorithms. Users can specify an algorithm based on a feature vector, the spatial proximity of features, k-d trees, and other techniques. Approximate-optimization algorithms generally have a lower time complexity than brute-force algorithms, and an approximate optimal matching result is often acceptable.

Feature Tracking

The tracking of matched features is essential to understanding the behavior of a data set. The framework allows the user to track features through two different methods:

The first tracking approach follows features over time using a *location-based search* (Banesh et al. 2017). For each reference feature detected in time step T_1, we search in T_2 for all features within a radius r of the reference. We declare features a match when the pair's separation is less than r. For a reference feature in T_1, if there is only one match in T_2, we consider the feature to have simply migrated from one position to another. However, if we find multiple matched features in T_2, we consider the reference feature split into two or more new features in the latter time step. The reverse is also true, and is considered a merge of multiple features into one. If we find no features within the radius r in T_2, we consider the feature to have died. Figure 6 below outlines some of these cases with further examples in Case Study II. We generate a bi-directional list of all such relations and

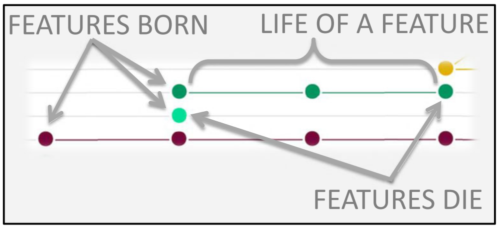


Fig. 6 Events in a feature tracking graph. Closeup view of a tracking graph. The first three types of tracking events are highlighted: feature birth, feature death, and a feature moving from one location to another with no changes

update the list with each new time step explored by the user. Any change in parameters other than time will update the entire list at once. This allows our framework to work quickly when the user is temporally parsing through their data as we expect that once the scientist finalizes parameter settings, the focus will be on the temporal evolution of features. Figure 7 shows an example of a location-based search by tracking segments corresponding to the Agulhas eddies. "Tails" representing the segments' previous n time step locations, here $n = 3$, allows a user to comprehend feature movement more intuitively. When regions split, as in Fig. 7(b), the joined tails detail the features' evolution.

The second tracking method allows users to observe regions that contain *sets of objects*, specifically objects or regions that are abstractly defined by features such as SIFT, SURF, and FAST. Two tracking scenarios are enabled this method: temporal tracking and spatial tracking (understood as tracking through a set of images ordered by a parameter other than time or not ordered). In both scenarios, we select a region of interest to track defined by the corners of a rectangle. Using FLANN-based matching, we match features from the user-specified region to features in an image of interest. The rectangle corners are mapped to the target image using a *homography* (Forsyth and Ponce 2011), to ensure that the region in the target image is a properly transformed version of the region initially specified. If the algorithm identifies an acceptable homography, this step of tracking is a success. We define a homography's acceptability by various factors such as if the matched region resides inside the image boundaries, the lines defining a matched region do not overlap or cross, and if the homography parameters are reasonably valued. When using this method to track temporally, at every time step, we segment the region identified by the homography in the previous time step to use as the reference region for the subsequent time step. This allows us to incorporate non-affine deformations into the tracking algorithm and results in stronger temporal tracking. When tracking spatially, we add every newly matched region from each image we inspect to a list of reference regions. We compare future regions to each entry in the list so that we may account for minor variations; this results in a more robust algorithm. An example of temporal tracking is shown in Fig. 8.

Interactive Exploration and Quantitative Analysis

Each algorithm in the toolkit has associated parameters that define a final visualization. The user interface provided allows a user to modify these parameter values in real-time. Different combinations of algorithms and parameter value choices lead to unique insights depending on the domain scientist's objective and data. This aspect of the visualization framework is a significant benefit to a scientist's data processing and analysis workflow because it allows them to taper the results of the computer vision pipeline to specific features in the data. As described in Tikhonova et al. (2010a), interaction with 2D images provides a sense of realism to 2D image data.

Once the user finalizes the features described by the computer vision pipeline, various quantitative tools are available to evaluate the results. The framework supports the computation of statistical properties for features over time. Figure 9(a) shows one such example regarding edges corresponding to eddies from sequential time steps. In this example, in each time step from T_0 to T_{12}, edges corresponding to the Agulhas eddies are extracted and overlaid in T_{12}. The eddies' relational positions from one time step to another and their overall movement over time are plainly visible. Users can now select a region of interest and derive statistical properties: mean, standard deviation, and minimal and maximal values for features in this region, Fig. 9(b).

A second tool allows users to visualize relationships and changes in features over time using count and tracking graphs. For a set of features from time step T_0 to time step T_n, we combine matching and tracking algorithms to

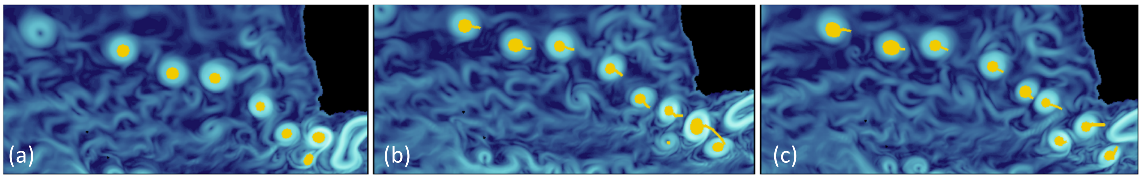


Fig. 7 Image segment tracking. Regions tracked from time step T_0 (a) to time step T_{14} (b) to time step T_{26} (c). Yellow tails indicate each feature's prior location. The split lines at the bottom-right of (b) show where a region from a previous step has split into two regions

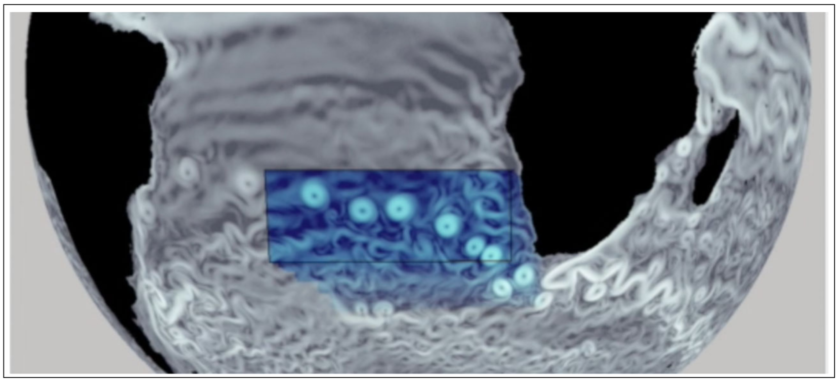


Fig. 8 A temporal tracking of the Agulhas rings. In this example, we track SURF features corresponding to the Agulhas eddies using FLANN-based matching. The selected region of interest, which was initially square, is now elongated and more rectangular as the eddies become more linear with respect to each other due to forces in the ocean. We vary the saturation to leave a trail of the locations tracked over the 27 time steps of the data in this example

find relationships between features over time. We map these relationships using a tracking graph. We can also map the total number of features found in each time step to a count graph. As graphs that count such features are traditionally used in various disciplines to monitor changes to dominant features over time, this addition supports a scientist's traditional analysis environment. We discuss examples of count and tracking graphs for eddy analysis in Case Study II, Section "Case Study II: Image Segmentation and Tracking for Analysis of Eddies in the Agulhas Region".

Finally, users can save the state of their results back into the Cinema database for post-processing and repeatability. One post-processing technique is the change detection algorithm introduced in Banesh et al. (2018, 2019). Feature-centric information such as feature count, the aforementioned statistical values or feature locations can be saved and used as input to the a change detection algorithm to identify important points in a sequence of paramtric values or through time.

Case Study Results and Discussion

The case studies presented highlight the robustness of the feature analysis toolkit, user interface, and quantitative techniques for various ocean science tasks. The studies were conducted in collaboration with ocean scientists to ensure applicability to a range of analysis problems. The Gulf Stream Case Study in Section "Case Study I: Edge Detection Supporting Analysis of the Gulf Stream" and the MPAS eddy analyses of Section "Case Study II: Image Segmentation and Tracking for Analysis of Eddies in the Agulhas Region" show the capability of the system for proxy images from MPAS-Ocean simulation results.

Case Study I: Edge Detection Supporting Analysis of the Gulf Stream

Subtropical ocean boundary currents, delineated by a coastline, transport water between the poles and the

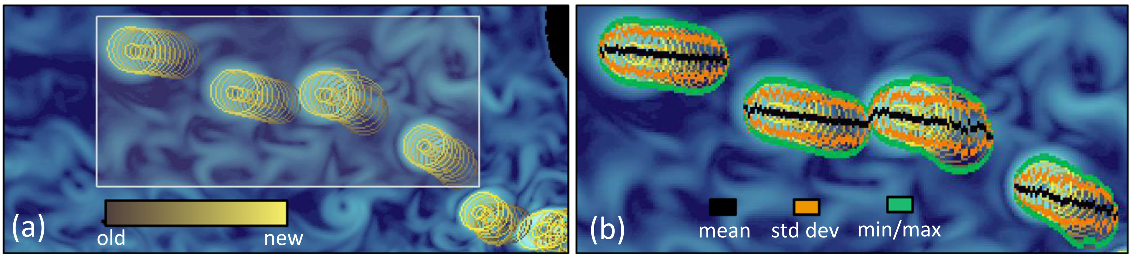


Fig. 9 A statistical analysis of image features. Edges in multiple time steps mapped to one image to visualize feature evolution (a). Edges are mapped with a yellow–brown colormap, with dark-to-light indicating older-to-newer edges. A user can select a region of interest (a) and compute statistical information (b) - mean (black), standard deviation (orange) and minimal/maximal values (green)

equator. Subtropical *western* boundary currents dominate the equator-to-pole heat transport and impact many aspects of ocean science. Western boundary currents and their associated temperature fronts correspond to regions of intensified atmosphere-ocean heat exchange (Kelly et al. 2010) and guide atmospheric storms (Hoskins and Hodges 2002). Scientists study the latitude at which these currents diverge from the shore and their subsequent pathways. These locations exhibit high gradients in temperature, with colder temperatures pole-ward and warmer temperatures equator-ward of the current.

The Gulf Stream, one such subtropical western boundary current, starts at the Gulf of Mexico, travels northward along the shoreline of the U.S. East Coast, and diverges from the coast at Cape Hatteras, NC. At this divergence, the warm water south of the Gulf Stream contrasts with the colder water to the north. The intersection of these waters is referred to as the Northern Boundary of the Gulf Stream and is a significant feature in ocean study. Ocean scientists employ various techniques to identify the Northern Boundary. One conventional method is to identify the 15° C isotherm of the temperature field at 200 m depth (Seidov et al. 2019). A second method uses the 25-cm contour of sea surface height (Andres 2016). Analysis of sea surface height is appropriate because in the Earth's oceans, the geostrophic balance, a balance between the pressure gradients and the Coriolis forces, correlates the sea surface height contours to streamlines of barotropic flow. Therefore, regions with the strongest currents, as at the Northern Boundary of the Gulf Stream, can be identified and measured through specific sea surface height contours. Other methods for identifying the Northern Boundary include finding regions of high gradients in both sea surface height and temperature (Seidov et al. 2019). The locations of largest gradients in sea surface height are where the jet of the Gulf Stream moves the fastest; this typically correlates to the Northern Boundary. Similarly, the locations with the largest gradient in temperature indicate the intersection of the cold and warm waters at the Northern Boundary. In this case study, we consider three tasks for identifying and evaluating the Gulf Stream's Northern Boundary. We choose to conduct these evaluations on a Cinema image proxy database of the temperature field by identifying regions with the largest gradients. We highlight the region of interest in Fig. 10. We subsequently evaluate the results provided by the framework as compared to traditional Gulf Stream analysis methods to discuss the advantages and disadvantages of the various techniques.

Task 1: Finding the Gulf Stream Northern Boundary

Identifying the Northern Boundary of the Gulf Stream is an essential task in ocean science for many reasons. In addition to the broader scientific implications, verifying the Gulf Stream location is critical to validating an ocean model's accuracy. Talandier et al. (2014) note that in an ocean model when the Gulf Stream's divergence point at Cape Hatteras is not correctly reproduced, it can invalidate many aspects of the simulation for the entire North Atlantic Ocean region. Such discrepancies can lead to errors or bias in the scientists' conclusions.

Edge detection, implemented via the Canny operator in our feature analysis toolkit, is an efficient method for identifying regions with large gradients. Since the Northern Boundary can be defined as a region with large gradients in temperature, edge detection is an appropriate technique to identify the current's divergence point and subsequent pathway. We use a Cinema database of the temperature parameter of MPAS-Ocean with a Mercator projection and focus on the North Atlantic Ocean region. We combine a Gaussian blur filter (Section "Image Filters") with Canny edge detection (Section "Feature Detection") to determine the locations of large gradients in temperature and extract the Northern Boundary. This is accomplished by first smoothing the data with a Gaussian filter of a small window size of 3 × 3; we keep the window size small to prevent displacement of edges. We then apply Canny edge detection and narrow the results to the intersection of the cold (blue) and warm (red) waters. Figure 11 shows the results of this process at time $t = 0$ of the simulation.

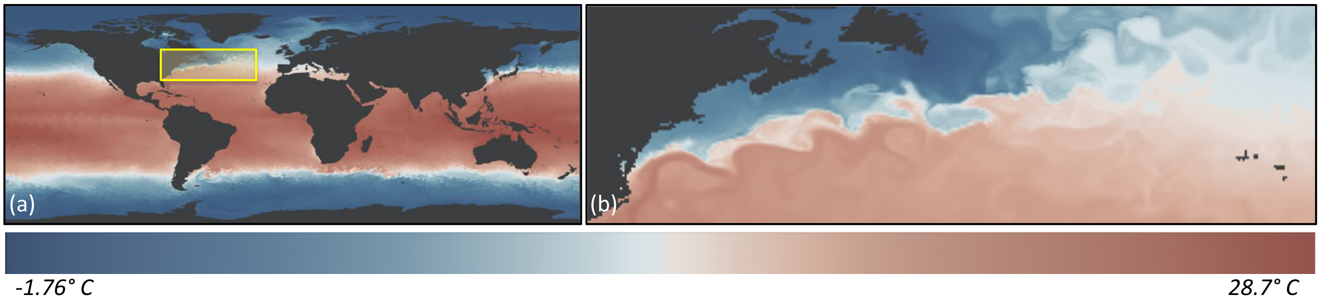


Fig. 10 MPAS-Ocean temperature field. (a) An image proxy of the Mercator projection of temperature in MPAS-Ocean using a muted red-blue colormap (cool/warm). (b) A close-up view of the yellow–boxed region shown in (a) highlights the Gulf Stream Northern Boundary

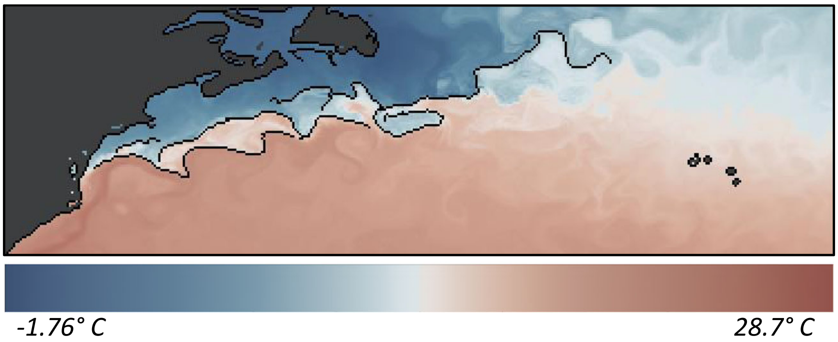


Fig. 11 Edge detection to identify the gulf stream. We apply a Gaussian blur filter with Canny edge detection, adjusting the parameters to extract the edges (black polylines) indicating large temperature gradient changes between hot and cold water along the Gulf Stream. Water temperature along this front ranges from 5 to 15 °C

The black polylines identify the locations of the largest gradients in the region. We see that these boundaries start almost exactly at the point of divergence, Cape Hatteras, and follow the Gulf Stream northeast towards Europe. Our collaborators confirmed that "The captured edges are related to the Gulf Stream 'North Wall' or Gulf Stream Landward Edge, which is typically derived from satellite data. By highly automating the extraction of the position of this edge from model data, we are able to provide rapid comparison to observations to assess a climate model's validity."

Task 2: Mapping the Gulf Stream Northern Boundary over Time

Temporal variations in the Gulf Stream's location can lead to an extensive impact on coastal cities. A recent study discusses how coastal cities located close to a western boundary current's temperature front experience drastic changes in regional climate due to minimal changes to the ocean current's position (Saba et al. 2016). Therefore, a system's ability to efficiently map regional variations in ocean current positions enhances the user's ability to assess regional climate changes due to changes in ocean circulation.

Our Cinema database of MPAS-Ocean temperature consists of 173 time steps in five-day intervals spanning 2.3 years. We extract the edges of the Northern Boundary for each time step through the process discussed in Task 1 and map all extracted edges to image T_{172} (starting from T_0). We render earlier time steps first and later time steps last. We use a yellow-to-brown colormap, where edges corresponding to the latest time steps are shown in bright-yellow tones and the earliest time steps in dark-brown tones.

Domain experts confirmed that in the results shown by the framework, the temperature front in areas closer to the coast shows less change over time and further away from the coast, exhibits significantly more temporal variability. The region of more compact flow is known as the Robust Zone, from Cape Hatteras to just east of the Grand Banks of Newfoundland, 75° West to 50° West longitude. Further out east is known as the Extension Zone, 50° West to 40° West longitude (Seidov et al. 2019). This breakpoint at 50° West longitude is precisely where the framework results start to show more temporal variability.

Furthermore, regions with a gradation from brown to yellow, as shown in Fig. 12(b), indicate that the Northern Boundary does not move sporadically in the Robust Zone but more gradually over time. This trend might correspond to the sinusoidal fluctuations of seasonal movement that has previously been observed of the Northern Boundary. In addition to seasonal variability, variations on shorter time scales might indicate eddy activity, while those over more extended periods may indicate changes in global ocean circulation.

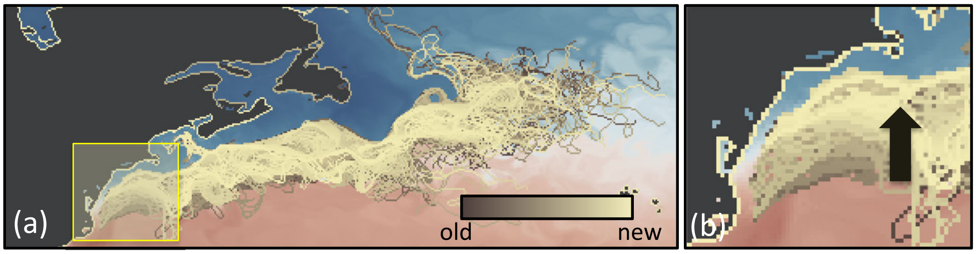


Fig. 12 A temporal analysis of the Gulf Stream. Mapping all 173 time steps of the MPAS simulation (a) allows scientists to view shifts in the temperature front of the Gulf Stream over time. For example, focusing on the region boxed in yellow, and enlarged in (b), the left-most region of the temperature front of the Gulf Stream shows a gradual shift northward (as indicated by the arrow)

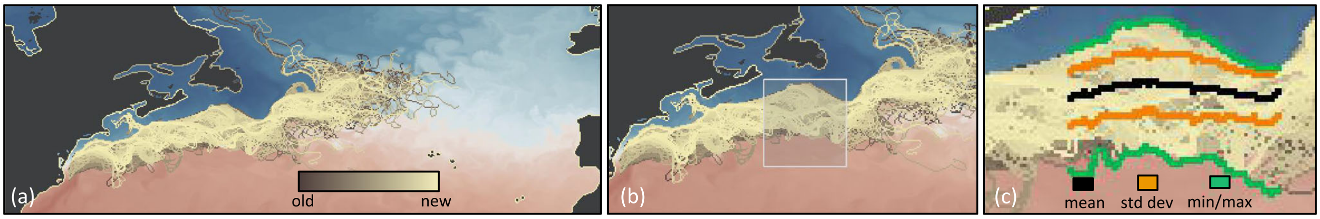


Fig. 13 A statistical analysis of the Gulf Stream. Using the temperature data set, we overlay edges from time step 0 to the current time step (a). Once edges have been derived, the user selects a region of interest (b) to derive statistical properties (c) based on the selected edges. For each longitudinal x-value in the selected region, we extract the mean (in black), the standard deviation (in orange) and the extreme values (in green)

Task 3: Quantitative Changes in the Gulf Stream Northern Boundary

Ocean scientists also derive statistical information from time-dependent data as observations and results from models can be noisy. Such time-averaged data is useful for comparisons or to highlight trends. We accomplish this analysis by selecting a subset of the edges extracted in Task 2 through a "click-and-drag" to select a rectangular region of interest, Fig. 13(b). For each longitudinal point within this selected region, we examine each edge pixel detected to compute mean, standard deviation, and extremal values. The results of these computations are overlaid on the edges, as shown in Fig. 13(c). Deriving such statistics is a standard method for Gulf Stream analysis within the ocean science community (Fig. 8 in Bryan et al. (2007), Fig. 2 in Talandier et al. (2014), and Figs. 3, 4, and 6 in Auer (1987)). The availability of such a technique allows ocean scientists to replication traditional ocean science analyses.

Evaluation

Traditional studies of the Gulf Stream examine both the temperature and sea surface height parameters to identify the Northern Boundary of the Gulf Stream. To evaluate the validity of image proxies for analysis of the Gulf Stream, we compare results from the image analysis framework to features from the original simulation data visualized using Paraview (Ahrens et al. 2005). We arrange the comparisons in the format shown in Fig. 14: The top row examines features extracted from the simulation in Paraview and the bottom row features from our presented framework. The left column examines the temperature parameter and the right column examines sea surface height. We establish colormaps for the temperature fields using a divergent hot/cold colormap and sea surface height data using a blue-toned gradient colormap. In all cases, we extract the features from the underlying floating-point data, not the colormapped information.

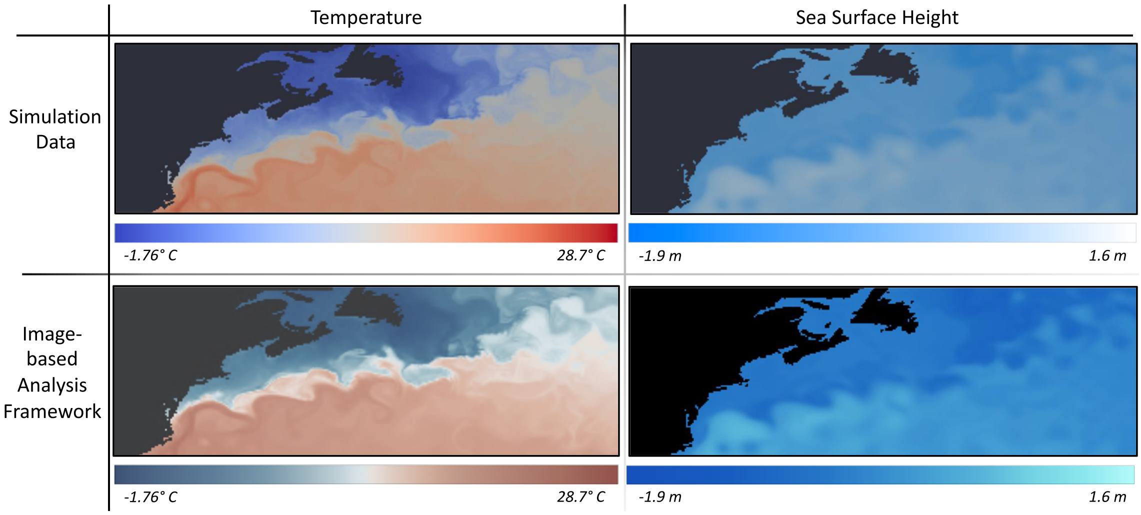


Fig. 14 A visual validation of proxy images. Temperature (left column) and sea surface height (right column) derived from the simulation and visualized in Paraview (top row) and by the image-based analysis framework (bottom row) show very similar features of the Gulf Stream

First, we examine the image proxies themselves to examine if they capture the features as represented in the simulation results, Fig. 14. We see that in the image proxies, despite converting the data from a Voronoi mesh grid to an image, the features of the Northern Boundary are captured clearly. Therefore, we expect future evaluations of these images to be a faithful representation of the original simulation data.

A Comparison of Gradient-based methods for Temperature and Sea Surface Height Our first study identifies locations of large gradients in both sea surface height and temperature. We know this is a method traditionally used to identify the location of the Northern Boundary. To extract these features from the original simulation data, we compute the gradient magnitude directly on the Voronoi grid by determining the derivatives in the latitudinal and longitudinal directions. These results are shown in the top row of Fig. 15. The gradient magnitude is colormapped with a grayscale mapping where regions with high magnitude are white and those with low magnitude are black. We extract similar features with the image-based framework using Canny edge detection and mark high-gradient regions with black poly-lines. These results are shown in the bottom row of Fig. 15.

An evaluation shows several differences in the results captured by the two techniques: In the simulation results, for both temperature and sea surface height, the high gradient regions are in sharp contrast to the background. However, the locations of the Northern Boundary are not defined by a single gradient magnitude but a range of values. Therefore, to extract the feature of interest, the scientist must threshold the results using a threshold range wide enough to capture the entire Northern Boundary but small enough to minimize the observed region. This uncertainty will result in wider regions identified rather than a sharp line. In contrast, the non-maximum suppression and hysteresis thresholding in the Canny detection algorithm defines sharp, continuous lines that properly link the region's largest gradients. This difference is especially apparent in the sea surface height field as large gradients of the Northern Boundary cover larger areas of the region. Additionally, the turbulence, eddies, and other noisy features in the sea surface height also have high-gradient magnitudes and might obfuscate the Northern Boundary's precise location. This feature is much more clearly defined by the image-based analysis framework, due to the addition of the Gaussian filter to remove this extraneous noise. Therefore, we observe that though the Canny algorithm uses the same gradient-based technique as the traditional approach, the refinements included in the former lead to a more effective identification of the specific features.

A Comparison of Gradient-Based Analysis to Other Methods We conduct a similar evaluation to compare the Canny gradient-based results to other typical ocean science techniques to analyze the Northern Boundary. As mentioned before, the integration of traditional ocean science analysis tools within the framework makes such comparisons straightforward and effortless. In Fig. 16, we compare the results from Canny edge detection to isovalue-based results from the simulation data. For temperature, we compare the

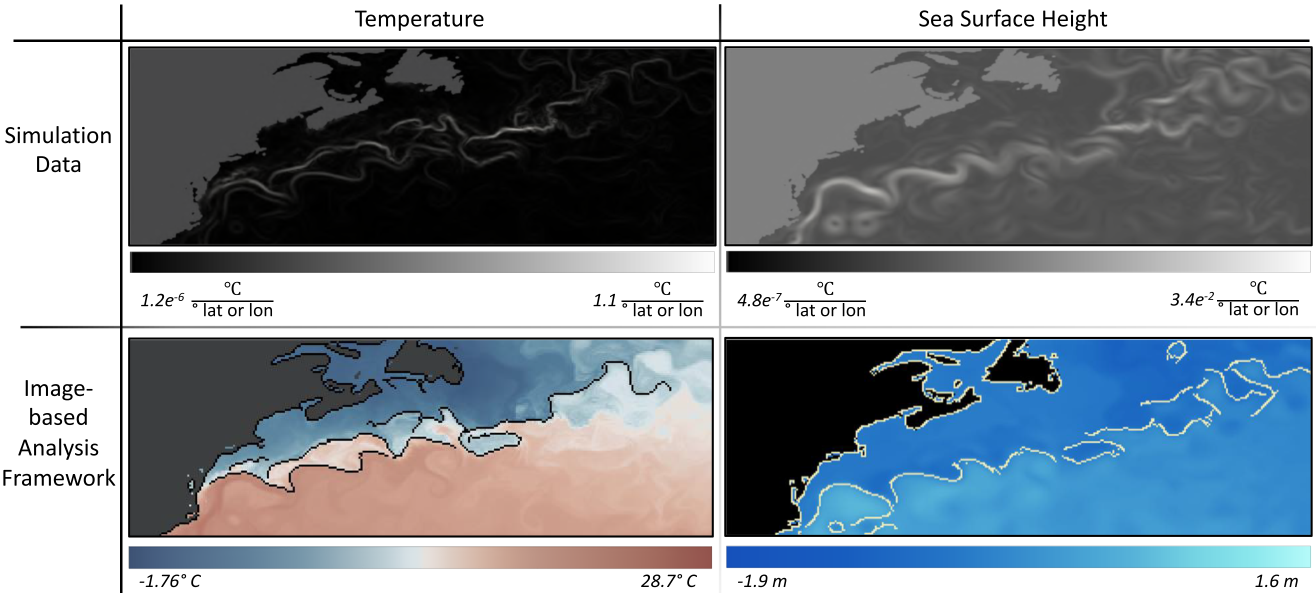


Fig. 15 Comparing gradient-based methods for Gulf Stream identification. We compare gradient-based methods, visualized using Paraview (Kitware 2012) (top row) and the image analysis framework (bottom row) to extract features related to the Gulf Stream

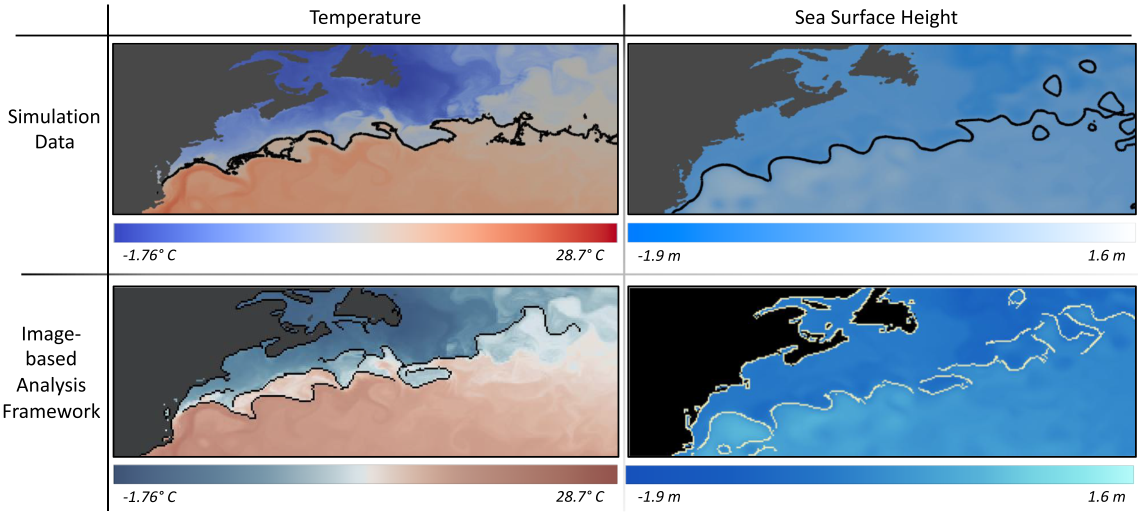


Fig. 16 Comparing edge detection to other techniques. We compare contours of temperature and sea surface height with edges defined by Canny edge detection within the framework. The algorithms identify similar locations for the Northern Boundary in the Robust Zone but results different in the Extension Zone

15°C isotherm of the simulation, mapped in Paraview with black poly-lines, to Canny edge detection results. For sea surface height, we compare the contours at −0.25 m, the corresponding value to 25 cm from observational data, to the results of Canny edge detection. For both parameters, the techniques' results are well correlated within the Robust Zone and less so in the Extension Zone. Within the Robust Zone, the poly-lines for each of the parameters of the two techniques overlap in multiple locations. However, in the Extension Zone, we see the isotherm contour shift southward while the corresponding Canny edge follows a northward trend. Similarly, in sea surface height, the isovalue contour becomes disorderly while the Canny edge results follow a strong northerly path. One additional difference is that both isovalue-based results identify many additional small, enclosed poly-lines that are unlikely to correspond to the Northern Boundary. In contrast, our gradient-based method provides a cleaner response.

We extend this comparison by extracting the statistics for these poly-line results over the 172 time steps of the simulation. Figure 17 shows the results of extracting the mean (in black), standard deviation (in orange) and maximum/minimum (in green) of each data set. A direct one-to-one comparison of these results does not necessarily match as we saw in the previous example that the results might vary. Regardless, it is important to note that scientists

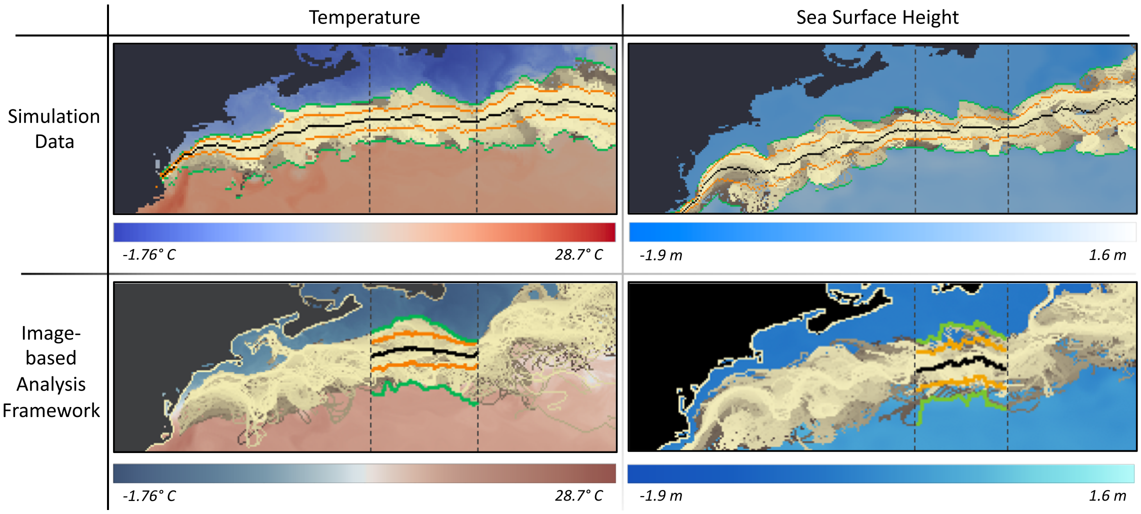


Fig. 17 A statistical comparison of the Northern boundary. For each Northern Boundary identification technique, we extract statistical information about the variations in locations over the 172 time steps of the simulation. The mean is shown in black, the standard deviation in orange and the minimum and maximum extremes in green

can indeed conduct such comparisons if needed. We see that, overall, the statistical trends do follow similar paths between the two analysis techniques. However, there is slightly more variability in the edge detection results than the isovalue-based results, so the minima and maxima encompass a broader range of latitudes.

Discussion

Our climate science collaborators value the intuitive usability, accuracy and speed of the presented tool compared to traditional statistical analysis techniques. The tool's flexibility in assessing the data, especially temporal data, from the underlying MPAS simulation allows the scientist to extract precise information for the northern Gulf Stream temperature front in much less time than when working with the high-dimensional data. One collaborator remarked that "The edge detection method is a high-quality method to quantify the location of the Gulf Stream." Another noted that "The resultant analysis visualizations were easy to compare and contrast with images from other publications, simulations and experimental data to assess simulation accuracy." Ocean scientists also noted that determining a current's boundary generally requires tedious and time-consuming aggregation and organization of experimental, satellite, or simulation data. For example, on average, loading *each* time step of MPAS-Ocean on a personal laptop in Paraview takes about 15 s. Loading the same information in the image-based framework is instantaneous due to the smaller data footprint. Additionally, as Cinema and the framework are not simulation-specific, the techniques are easily applied to other ocean models and data sets. It is clear from this case study that the presented framework makes it possible to quickly and effectively accomplish analysis tasks for the ocean sciences.

Case Study II: Image Segmentation and Tracking for Analysis of Eddies in the Agulhas Region

Mesoscale eddies are large vortices in the oceans, typically 10 to 150 km in diameter (Woodring et al. 2015), with widespread impact. They influence the ocean's biological network (Chelton et al. 2011), can contribute to heat transport over hundreds of miles (Volkov et al. 2008), affect weather conditions in the ocean and impact various other aspects of ocean dynamics (McWilliams 2008). Eddies can also play a significant role in transporting pollutants from human-related disasters, for example, during the Fukushima Nuclear Power Plant accident (Budyansky et al. 2015) and the Deep Water Horizon oil spill (Walker et al. 2011). These large rotating bodies of water play a role in ocean processes on a broad range of space and time scales, and are significant to various aspects of ocean science.

Ocean scientists utilize a wide range of methods for eddy detection. One common technique is to extract threshold-based segments of the Okubo-Weiss metric (Williams et al. 2011a; Williams et al. 2011b; Petersen et al. 2013). Okubo-Weiss measures rotational behavior in a turbulent fluid and identifies features such as eddies with high rotational energy. A second geometric approach for eddy detection identifies threshold-based segments of the sea surface height parameter (Chelton et al. 2011). Wavelet analysis has also been an efficient approach for identifying these localized coherent structures (Doglioli et al. 2007). In this case study, we consider two main tasks for the study of mesoscale ocean eddies that form at the southern tip of Africa. The Agulhas Current, an ocean current that follows the eastern boundary of Africa, pinches off eddies near Cape Town and inverts eastward as the Agulhas Return Current. Eddies that pinch off at this retroflection move westward across the Atlantic Ocean towards South America. We perform these eddy-related tasks using the feature analysis framework by identifying regions defined by a threshold value and tracking them using a location-based search. We evaluate the results provided by the framework through a comparison with conventional eddy detection techniques.

Task 1: Identifying the Agulhas Eddies

Image segmentation, a feature identification method in the image analysis framework, is an appropriate tool for detecting ocean eddies. We use a Cinema database of sea surface height of the MPAS-Ocean simulation and focus on the South Atlantic Ocean. We detect eddies in each image using a combination of image segmentation and image moment computation to constrain the size of features detected to those corresponding to eddies. We define image segments as regions of the image with values at or above a particular threshold. This culls out the background ocean and retains only salient features such as eddies, high-velocity currents and some ocean turbulence. We then refine the search further to extract eddies of interest using moments. Moments are weighted averages of the region's pixel intensities that describe various properties such as shape, size, location, and orientation (Flusser et al. 2016). We are particularly interested in the size of the segment, the 0th order element (M_{00}) of the moment matrix. We use this information, along with the minimum and maximum size thresholds to remove shapes that are too small (ocean turbulence) or too large (high-velocity currents). The location of the eddies, determined by its center of mass, is identified by the 0th (M_{00}) and 1st (M_{01}, M_{10}) order elements of the moment matrix:

$$x = M_{10}/M_{00}$$

$$y = M_{01}/M_{00}$$

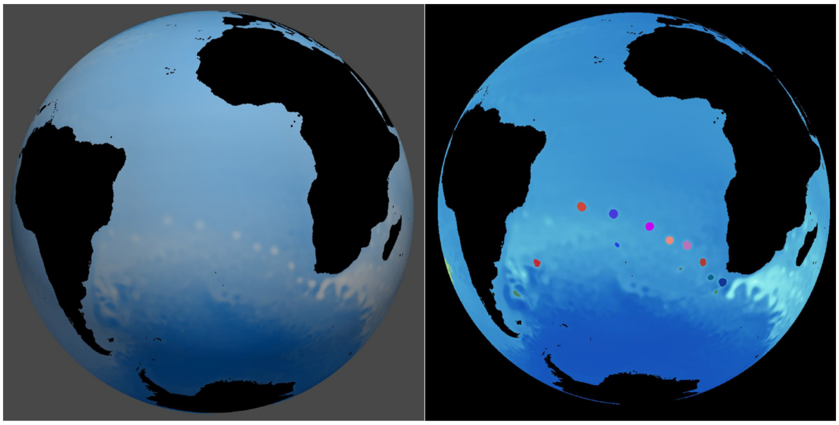


Fig. 18 Eddy detection using sea surface height. Using the sea surface height field, left, we can identify eddies in the South Atlantic Ocean using image segmentation, right

Features identified using this method are shown in Fig. 18 (right). Each new eddy segment is assigned a new random color, and the region filled with the corresponding color.

Task 2: A Temporal Analysis of the Agulhas Eddies

Mesoscale eddies, such as the Agulhas Eddies, also transport temperature and salinity (Beal et al. 2011; Hogg et al. 2011) across large distances and affect marine biology (Gaube et al. 2013) as they travel for weeks to over a year. We can track this temporal eddy movement through the image analysis framework. Here, we track eddies over the 173 time steps of this particular MPAS-Ocean simulation using a location-based search and setting the search radius to 25 pixels, a value deemed appropriate by our ocean scientist collaborators. As eddies move through the image sequence, we capture this migration in our framework by a 'tail' connecting their last five locations, Fig. 19. By studying the trail behavior, we can see how fast eddies are moving across the ocean, the merging and splitting of eddies, and the general progression of sets of eddies. As eddies die, their tails linger for a few more time steps, to inform users where past eddies were in relation to new eddies that might be approaching the same region. This visualization is especially important for eddies such as those from the Agulhas as they cross the Atlantic Ocean, losing kinetic energy on the way. Understanding regions where eddies are more likely to die helps ocean scientists better explain the region's oceanic events.

Scientists can further analyze feature tracking through the associated tracking graph. The tracking graph shows the births, lives, and deaths of eddies from time step zero to the current time step. We perform tracking using a location-based technique, as described in Section "Analysis Framework".

Figure 20 bottom shows the particulars of the Tracking Graph. Each unit along the x-axis represents a time step, and each row shows the life of a particular eddy. At any particular time step, there are five cases to consider: the birth of an eddy, the death of an eddy, one eddy moving to another location, one eddy splitting into multiple pieces with each of those pieces moving, and multiple eddies converging into one eddy. When an eddy is born and assigned a random color, we place a dot of the same color on the Eddy Tracking Graph at the current time step and on a new row of the graph. If the eddy is only alive for that one time step, the dot will remain, but there will be no connecting lines. If the eddy lives for more than one time step, there will be lines connecting the dot of its birth to the last dot of its death.

The two unusual cases arise when one eddy splits into many eddies or when many eddies merge into a few. Figure 21 illustrates one of these cases, when one eddy splits into two. This action is represented in the tracking graph by a branching of the tracking line to form two tracks of the

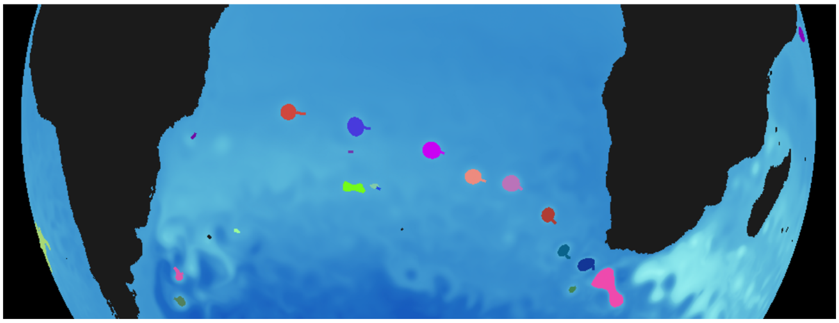


Fig. 19 Eddy tails in temporal tracking. Visualizing eddy tails as these features move across the Atlantic Ocean shows their temporal progression. This figure shows time step 9 of the MPAS-Ocean simulation

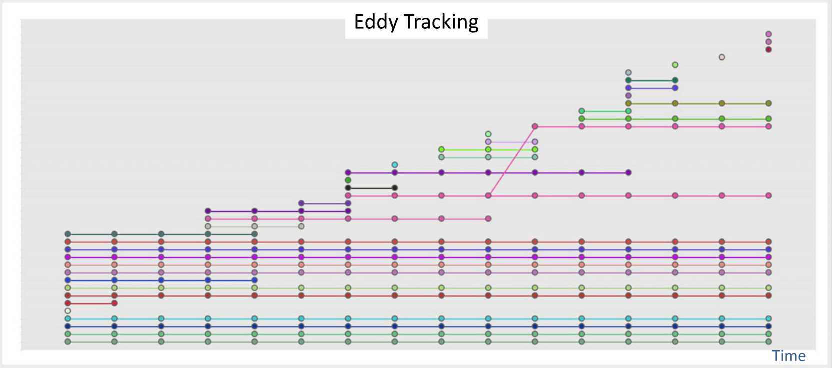


Fig. 20 Eddy Tracking Graph. Image segments that correspond to eddies can be tracked through life events such as births, deaths, splits and merges using a tracking graph

same color moving forward in time. In cases where many eddies merge into one, lines are drawn from the last point of the individual eddies to an entirely new eddy path on the graph. When many eddies merge to form one, the new eddy is not interpreted as the continuation of any old eddy but viewed it as an entirely new one. The particular pink eddy of interest in Fig. 21 starts to split around time $t = 9$, splits completely into two eddies at $t = 10$ and is seen moving across the Atlantic Ocean at later time steps. Eddies that split from their parents are given the same colors to denote their lineage. These eddies' tails also connect, showing the user that they both originate from the same parent.

Evaluation

Standard eddy detection methods use either sea surface height or the Okubo-Weiss parameter to identify eddy features. We compare the results for eddy analysis in sea surface height identified by the framework to corresponding results from the simulation data visualized in Paraview (Ahrens et al. 2005). To extract features corresponding to eddies in Paraview, we set a threshold on sea surface height from 0.3 to 2 to closely identify the corresponding regions. Collaborating climate scientists identified these values as appropriate to how this data would traditionally be analyzed. Figure 22 middle shows the results of this thresholding. A comparison of these segments to those from the image-based framework, Fig. 22 right, shows that both methods identically segment a large number of eddies, specifically the 11 major segments that correspond to the trans-Atlantic eddies. In addition, both methods also identify the single eddy segment near the southern tip of South America. However, in the simulation data, because we place no size restrictions on the results, larger portions of sea surface height are also selected through they do not correspond to eddies. In contrast, our image-based framework results do not suffer from this issue due to size constraints on the results. This also allows the framework to extract one more eddy, notably the bright red eddy close to South America, which was lost in large masses extracted from the simulation data.

A Comparison of Sea Surface Height to Okubo-Weiss We also compare features identified by sea surface height to those identified by the Okubo-Weiss parameter to determine which parameter might be better suited for eddy detection. Figure 23 middle and right show the two thresholds that must be used to identify eddies in the Agulhas region using sea surface height, to capture the entirety of the eddies that exist on both the left and right side of Cape Town, South Africa. This is also true of the results produced by our image-based analysis framework, Fig. 24, though our results are more accurate than those derived through the traditional Paraview method. However, identifying the eddies using Okubo-Weiss requires only need a single threshold. Adversely, Okubo-Weiss also

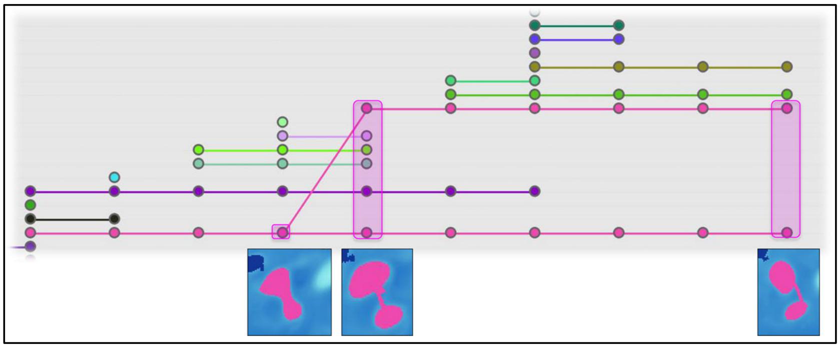


Fig. 21 Eddy splits. We track the pink eddy of interest as an example of how eddy splits are handled within the framework. The tracking graph shows the addition of a new line when the new eddy is formed. The eddy tails are connected to show lineage

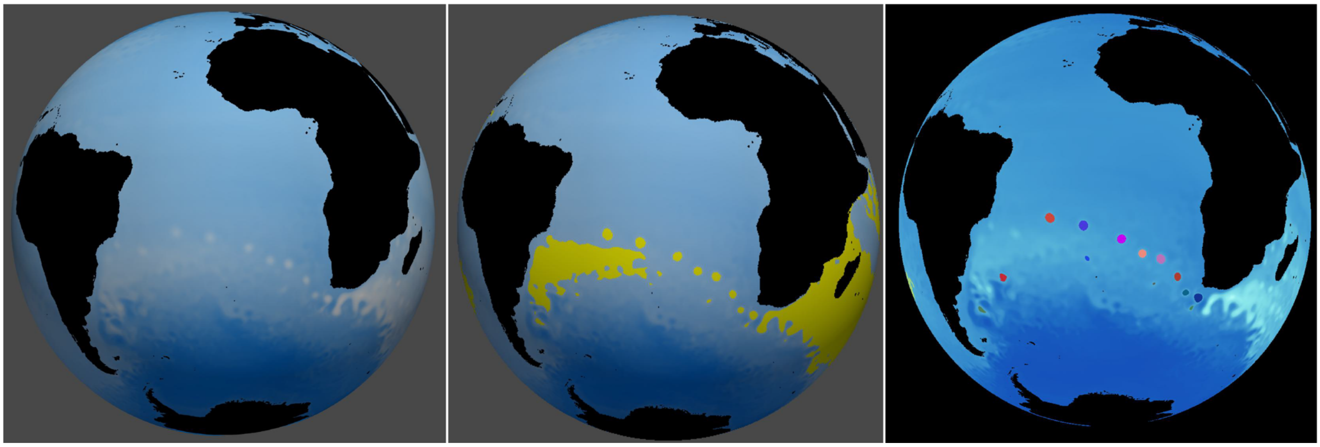


Fig. 22 Eddy detection using sea surface height. For sea surface height, left, we evaluate two methods for eddy detection. The first, shown in the middle, uses a threshold range to identify eddies from the simulation data using Paraview. The results are shown in bright yellow. The second, shown on the right, uses the image-based framework to identify segments and refine these segments by size. The extra refinement results in a more accurate eddy detection

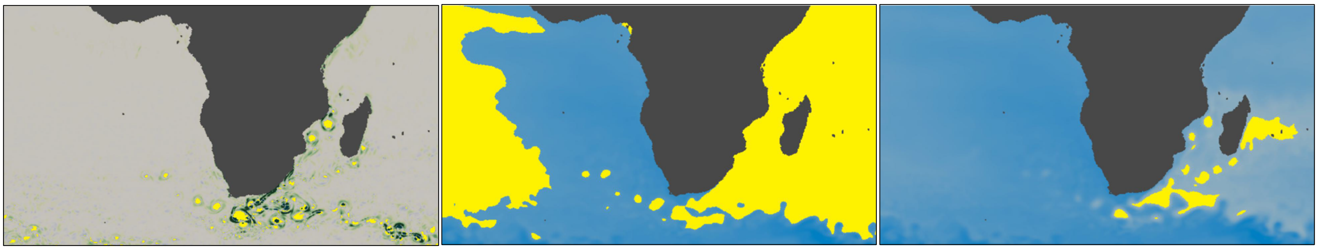


Fig. 23 Simulation-derived features for Eddy detection. To identify eddies in Agulhas region using sea surface height in Paraview, two sets of thresholds must be selected to capture the entirety of the eddies. However, only a single threshold range is needed when evaluating Okubo-Weiss

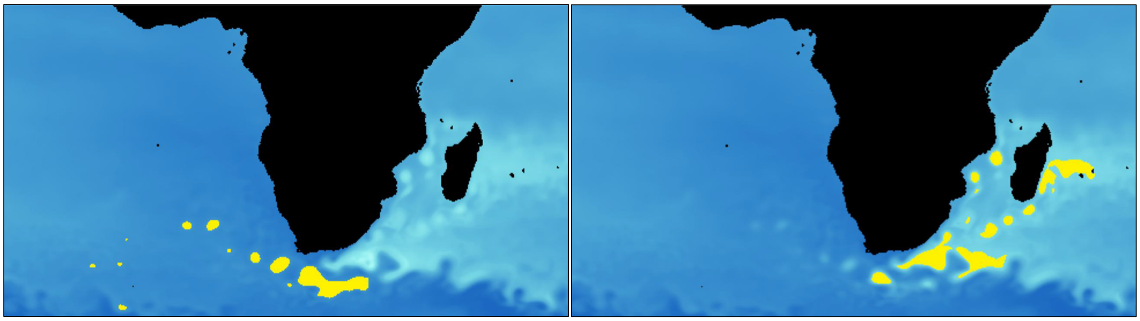


Fig. 24 Image segmentation for Eddy detection in sea surface height. Comparing the image-based results to the simulation results for sea surface height shows similar results. The simulation still identifies large, unrelated segments that are not present in our image-based results, but the smaller, eddy regions correlate between the two methods

identifies a substantial number of smaller features that correspond to turbulence in the ocean rather than eddy features. We will need to remove these smaller segments in post-processing. Nevertheless, it is much easier to remove extraneous features that correspond to noise than merging multiple sets of results. Therefore, scientists might find better results in conducting future evaluations of eddies using the Okubo-Weiss criterion rather than sea surface height when possible. This conclusion is not a fault of the image framework but a limitation of the sea surface height field as a means to identify eddies.

Discussion

As numerical simulations of the global ocean are using increasingly higher resolutions in space and time, they produce ever-larger amounts of data, ranging from several gigabytes to terabytes (Woodring et al. 2015). Producing information, such as Okubo-Weiss, from such simulated data sets can require additional hours of data processing time when using traditional approaches. Many commonly used eddy analysis tools (Chelton et al. 2011; Williams et al. 2011a; Williams et al. 2011b) are labor- or time-intensive and often limit the number of time steps, or the extent of the simulation domain studied, due to the size of the simulated data. There is a need for robust and efficient tools to obtain the desired and valuable information captured in such simulations. The use of Cinema to minimize the data size and the application of image-based techniques help improve on traditional methods for eddy detection. One ocean scientist collaborator commented that “This eddy detection and tracking software is a practical and computationally efficient tool for extracting important details of ocean eddy statistics. The graphical display of results is both intuitive and incredibly informative.”

Conclusion and Future Work

We have presented an exploratory visualization framework with a feature analysis toolkit and an interactive user interface. As demonstrated through science-relevant case studies, the framework’s flexible component design makes it possible to answer important scientific questions for simulation and experimental data efficiently. It is planned to expand the framework by incorporating machine-learning methods, in support of additional analysis tasks. The presented case studies document that our framework allows users to gain scientifically relevant insights into their data through 2D image proxies that users can analyze with a laptop.

Funding The authors thank David Rogers for his support and input. D. Banesh and T.L. Turton were supported by the Cinema project as part of the Exascale Computing Project. M. Petersen was supported as part of the Energy Exascale Earth System Model (E3SM) project, funded by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Declarations

All ethical standards have been followed to the best of the authors’ knowledge and abilities. The authors confirm that the manuscript has not be submitted simultaneously to any other publication. The authors also confirm that the vast majority (85%) of this work has not been published elsewhere and that this manuscript is a significant expansion of previous work. The previous work has also been properly cited in the manuscript. The authors also state that the results of their work have been presented as clearly and honestly as possible, with no fabrication, falsification or inappropriate data manipulation. In addition, all work presented are the authors’ own and references to related work has been properly cited with permissions obtained.

Ethical approval There were no human or animal studies that were a part of this work. Not Applicable.

Informed consent There were no human or animal studies that were a part of this work. Not Applicable.

Conflict of interest The authors declare no competing interests.

References

Ahrens J, Geveci B, Law C (2005) Paraview: An end-user tool for large data visualization. The visualization handbook 717

Ahrens J, Jourdain S, O’Leary P, Patchett J, Rogers D, Petersen M (2015) An image-based approach to extreme scale in situ visualization and analysis. International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015:424–434. https://doi.org/10.1109/SC.2014.40

Aigner W, Miksch S, Schumann H, Tominski C (2011) Visualization of time-oriented data. Springer Science & Business Media

Andres M (2016) On the recent destabilization of the Gulf Stream path downstream of Cape Hatteras. Geophys Res Lett 43(18):9836–9842

Auer SJ (1987) Five-year climatological survey of the Gulf Stream system and its associated rings. J Geophys Res: Oceans 92(C11):11709–11726

Banesh D, Schoonover JA, Ahrens JP, Hamann B (2017) Extracting, visualizing and tracking mesoscale ocean eddies in two-dimensional image sequences using contours and moments. In: Rink K, Middel A, Zeckzer D, Bujack R (eds) Workshop on Visualisation in Environmental Sciences (EnvirVis), The Eurographics Association

Banesh D, Wendelberger J, Petersen M, Ahrens J, Hamann B (2018) Change point detection for ocean eddy analysis. In: Proceedings of the workshop on visualisation in environmental sciences, eurographics association, Brno, Czech Republic, EnvirVis '18, pp 27–33

Banesh D, Petersen M, Wendelberger J, Ahrens J, Hamann B (2019) Comparison of piecewise linear change point detection with traditional analytical methods for ocean and climate data. Environmental Earth Sciences: 78. https://doi.org/10.1007/s12665-019-8636-y

Bauer A, Abbasi H, Ahrens J, Childs H, Geveci B, Klasky S, Moreland K, O'Leary P, Vishwanath V, Whitlock B, Bethel EW (2016) In situ methods, infrastructures, and applications on high performance computing platforms. Computer Graphics Forum 35:577–597. https://doi.org/10.1111/cgf.12930

Bay H, Tuytelaars T, V Gool L (2006) SURF: Speeded up robust features. In: Computer Vision and Image Understanding - CVIU, vol 110, pp 404-417

Beal LM, De Ruijter WPM, Biastoch A, Zahn R et al (2011) On the role of the Agulhas system in ocean circulation and climate. Nature 472(7344):429–436

Bradski G (2000) The OpenCV Library. Dr Dobb's Journal of Software Tools

Bryan FO, Hecht MW, Smith RD (2007) Resolution convergence and sensitivity studies with North Atlantic circulation models. part i: The western boundary current system. Ocean Model 16(3):141–159

Budyansky MV, Goryachev VA, Kaplunenko DD, Lobanov VB, Prants SV, Sergeev AF, Shlyk NV, Uleysky MY (2015) Role of mesoscale eddies in transport of Fukushima-derived cesium isotopes in the ocean. Deep-Sea Res I Oceanogr Res Pap 96:15–27

Caldwell PM, Mametjanov A, Tang Q, Van Roekel LP, Golaz JC, Lin W, Bader DC, Keen ND, Feng Y, Jacob R, Maltrud ME, Roberts AF (2019) The DOE E3SM Coupled Model Version 1: Description and results at high resolution. J Adv Model Earth Sys 11(12):4095–4146. https://doi.org/10.1029/2019MS001870

Canny J (1986) A computational approach to edge detection. Pattern Anal Mach Intell. IEEE Transactions on PAMI-8: 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Chelton DB, Gaube P, Schlax MG, Early JJ, Samelson RM (2011) The influence of nonlinear mesoscale eddies on near-surface oceanic chlorophyll. Science 334(6054):328–332

Childs H, et al. (2012) VisIt: an end-user tool for visualizing and analyzing very large data. In: High performance visualization—enabling extreme-scale scientific insight. CRC Press/Francis–Taylor Group, pp 357–372

COSIM Group at LANL & NCAR (2019) Mpas-Model. https://github.com/MPAS-Dev/MPAS-Model/

Doglioli AM, Blanke B, Speich S, Lapeyre G (2007) Tracking coherent structures in a regional ocean model with wavelet analysis: Application to Cape Basin eddies. J Geophys Res Oceans 112 (C5)

DSS Group at LANL (2021) Welcome to cinema science. https://cinemascience.github.io/

Dutta S, Biswas A, Ahrens J (2019) Multivariate pointwise information-driven data sampling and visualization. Entropy 21(7):699

E3SM Project (2018) Energy Exascale Earth System Model (E3SM). [Computer Software] https://doi.org/10.11578/E3SM/dc.20180418.36

Flusser J, Zitova B, Suk T (2016) 2D and 3D image analysis by moments. Wiley, Hoboken

Forsyth DA, Ponce J (2011) Computer vision: a modern approach. Pearson

Gaube P, Chelton DB, Strutton PG, Behrenfeld MJ (2013) Satellite observations of chlorophyll, phytoplankton biomass, and ekman pumping in nonlinear mesoscale eddies. J Geophys Res Oceans 118(12):6349–6370

Golaz JC, Caldwell PM, Van Roekel LP, Petersen MR et al (2019) The DOE E3SM coupled model version 1: overview and evaluation at standard resolution. J Adv Model Earth Sys 11(7):2089–2129. https://doi.org/10.1029/2018MS001603

Hogg AM, Dewar WK, Berloff P, Ward ML (2011) Kelvin wave hydraulic control induced by interactions between vortices and topography. J Fluid Mech 687:194–208

Hoskins BJ, Hodges KI (2002) New perspectives on the Northern Hemisphere winter storm tracks. J Atmospheric Sci 59(6):1041–1061

Hu M (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8(2):179–187

Johnson HJ, McCormick M, Ibáñez L (2013) Consortium TIS. The ITK Software Guide. Kitware, Inc., 3rd edn

Kageyama A, Yamada T (2014) An approach to exascale visualization: interactive viewing of in-situ visualization. Comput Phys Commun 185(1):79–85

Kelly KA, Small RJ, Samelson RM, Qiu B, Joyce TM, Kwon Y, Cronin MF (2010) Western boundary currents and frontal air–sea interaction: Gulf Stream and Kuroshio Extension. J Clim 23(21):5644–5667

Kitware (2012) In situ — paraview. http://www.paraview.org/in-situ/

Li S, Marsaglia N, Garth C, Woodring J, Clyne J, Childs H (2018) Data reduction techniques for simulation, visualization and data analysis. In: Computer Graphics Forum, Wiley Online Library, vol 37, pp 422-447

Liu L, Ozer S, Bemis K, Takle J, Silver D (2013) An interactive method for activity detection visualization. In: 2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV). IEEE, pp 129-130

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

Lukasczyk J, Weber G, Maciejewski R, Garth C, Leitte H (2017) Nested tracking graphs. In: Computer Graphics Forum, Wiley Online Library, vol 36, pp 12–22

McWilliams JC (2008) The nature and consequences of oceanic eddies. Ocean Modeling in an Eddying Regime: 5–15

Muelder C, Ma K (2009) Interactive feature extraction and tracking by utilizing region coherency. In: Visualization symposium, 2009. PacificVis' 09. IEEE Pacific, IEEE, pp 17-24

Muja M, Lowe D (2009) Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1) 2(331–340): 2

Ozer S, Silver D, Bemis K, Martin P (2014) Activity detection in scientific visualization. IEEE Trans Vis Comput Graph 20(3):377–390

Petersen MR, Williams SJ, Maltrud ME, Hecht MW, Hamann B (2013) A three-dimensional eddy census of a high-resolution global ocean simulation. J Geophys Res Oceans 118(4):1759–1774

Petersen MR, Jacobsen DW, Ringler TD, Hecht MW, Maltrud ME (2015) Evaluation of the arbitrary Lagrangian-Eulerian vertical coordinate method in the MPAS-Ocean model. Ocean Model 86(0):93–113. https://doi.org/10.1016/j.ocemod.2014.12.004

Petersen MR, Asay-Davis XS, Berres AS, Chen Q, Feige N, Hoffman MJ, Jacobsen DW, Jones PW, Maltrud ME, Price SF, Ringler TD, Streletz GJ, Turner AK, Van Roekel LP, Veneziani M, Wolfe JD, Wolfram PJ, Woodring JL (2019) An evaluation of the ocean and sea ice climate of E3SM using MPAS and interannual CORE-II forcing. J Adv Model Earth Sys 11(5):1438–1458. https://doi.org/10.1029/2018MS001373

Post FH, Vrolijk B, Hauser H, Laramee RS, Doleisch H (2003) The state of the art in flow visualisation: feature extraction and tracking. In: Computer Graphics Forum, Wiley Online Library, vol 22, pp 775–792

Reinders F, Post F, Spoelder HJW (2001) Visualization of time-dependent data with feature tracking and event detection. Vis Comput 17(1):55–71

Ringler T, Petersen M, Higdon RL, Jacobsen D, Jones PW, Maltrud M (2013) A multi-resolution approach to global ocean modeling. Ocean Model 69(Supplement C):211–232. https://doi.org/10.1016/j.ocemod.2013.04.010

Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. In: European conference on computer vision. Springer, pp 430–443

Saba V, Griffies S, Anderson W, Winton M, Alexander M, Delworth T, Hare J, Harrison M, Rosati A, Vecchi G, et al. (2016) Enhanced warming of the Northwest Atlantic Ocean under climate change. J Geophys Res: Oceans 121(1):118–132

Samtaney R, Silver D, Zabusky N, Cao J (1994) Visualizing features and tracking their evolution. Computer 27(7):20–27

Schmidt J, Gröller ME, Bruckner S (2013) Vaico: Visual analysis for image comparison. IEEE Trans Vis Comput Graph 19(12):2090–2099

Schneider CA, Rasband WS, Eliceiri KW (2012) Nih image to imagej: 25 years of image analysis. Nature methods 9(7):671–675

Seidov D, Mishonov A, Reagan J, Parsons R (2019) Resilience of the Gulf Stream path on decadal and longer timescales. Sci Rep 9(1):1–9

Silver D, Wang X (1997) Tracking and visualizing turbulent 3D features. IEEE Trans Vis Comput Graph 3(2):129–141

Suzuki S et al (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graphics Image Process 30(1):32–46

Talandier C, Deshayes J, Treguier A, Capet X, Benshila R, Debreu L, Dussin R, Molines J, Madec G (2014) Improvements of simulated Western North Atlantic current system and impacts on the AMOC. Ocean Model 76:1–19

Tikhonova A, Correa CD, Ma K (2010a) An exploratory technique for coherent visualization of time-varying volume data, vol 29, pp 783–792

Tikhonova A, Correa CD, Ma K (2010b) Visualization by proxy: A novel framework for deferred interaction with volume data. IEEE Trans Vis Comput Graph 16(6):1551–1559

Tinevez J, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, Bednarek SY, Shorte SL, Eliceiri KW (2017) Trackmate: an open and extensible platform for single-particle tracking. Methods 115:80–90

Turton T, Banesh D, Overmyer T, Sims B, Rogers D (2020) Enabling domain expertise in scientific visualization with cinemascience. IEEE Comput Graph Appl 40(1):90–98. https://doi.org/10.1109/MCG.2019.2954171

Volkov DL, Lee T, Fu L (2008) Eddy-induced meridional heat transport in the ocean. Geophys Res Lett 35(20)

Walker NDND, Pilley CTCT, Raghunathan VVVV, D'Sa EJEJ, Leben RRRR, Hoffmann NGNG, Brickley PJPJ, Coholan PDPD, Sharma NN, Graber HCHC, et al. (2011) Impacts of loop current frontal cyclonic eddies and wind forcing on the 2010 Gulf of Mexico oil spill. Monitoring and modeling the Deepwater Horizon oil spill: a record-breaking enterprise, pp 103–116

Wang K, Shareef N, Shen H (2018) Image and distribution based volume rendering for large data sets. In: 2018 IEEE Pacific Visualization Symposium (PacificVis). IEEE, pp 26–35

Wang K, Wei T, Shareef N, Shen H (2019) Ray-based exploration of large time-varying volume data using per-ray proxy distributions. IEEE Trans Vis Comput Graph

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Weissenböck J, Amirkhanov A, Gröller E, Kastner J, Heinzl C (2016) PorosityAnalyzer: Visual analysis and evaluation of segmentation pipelines to determine the porosity in fiber-reinforced polymers. In: 2016 IEEE Conference on Visual Analytics Science and Technology (VAST). IEEE, pp 101–110

Widanagamaachchi W, Christensen C, Pascucci V, Bremer P (2012) Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In: IEEE symposium on large data analysis and visualization (LDAV). IEEE, pp 9-17

Williams S, Hecht M, Petersen M, Strelitz R, Maltrud M, Ahrens J, Hlawitschka M, Hamann B (2011a) Visualization and analysis of eddies in a global ocean simulation. In: Computer Graphics Forum, Wiley Online Library, vol 30, pp 991–1000

Williams S, Petersen M, Bremer P, Hecht M, Pascucci V, Ahrens J, Hlawitschka M, Hamann B (2011b) Adaptive extraction and quantification of geophysical vortices. IEEE Trans Vis Comput Graphics 17(12):2088–2095

Woodring J, Petersen M, SchmeiSSer A, Patchett J, Ahrens J, Hagen H (2015) In situ eddy analysis in a high-resolution ocean climate model. IEEE Trans Vis Comput Graphics 22(1):857–866. https://doi.org/10.1109/TVCG.2015.2467411

Wu J, Ma H, Zhou M (2007) Vorticity and vortex dynamics. Springer Science & Business Media

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.