Front and Skeleton Features Based Methods for Tracking Salinity Propagation in the Ocean

Upkar Singh[a], T.M. Dhipu[b], P. N. Vinayachandran[c] and Vijay Natarajan[a,*]

[a]Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India
[b]Centre for Airborne Systems, DRDO, Bangalore, India
[c]Centre for Atmospheric and Oceanic Sciences, Indian Institute of Science, Bangalore 560012,India

ARTICLE INFO

Keywords:
Bay of Bengal
High Salinity Core
Tracking
Skeleton
Surface Front
Visualization

ABSTRACT

The Bay of Bengal (BoB) fosters several monsoon depressions and cyclones, playing a crucial role in the Asian summer and winter monsoons. The capacity of the bay to remain warm and energize such weather systems is attributed to its strong vertical stratification sustained by the large freshwater input into the bay. River runoff and rainfall into the northern bay in contrast to the high salinity water intrusion in the south creates a strong north-south salinity gradient. Here, we present a visual analysis tool to trace the path of the high salinity core (HSC) entering into the BoB from the Arabian Sea. We introduce two feature definitions that represent the movement and shape of the HSC, and algorithms to track their evolution over time. The two feature representations, namely fronts and skeletons, are based on geometric and topological analysis of the HSC. The method is validated via comparison with well established observations on the flow of the HSC in the BoB, including its entry from the Arabian Sea and its movement near Sri Lanka. Further, the visual analysis and tracking framework enable new detailed observations on forking behavior near the centre of the BoB and subsequent northward movement of the HSC. The tools that we have developed offer new perspectives on the propagation of high salinity water and its mixing with the ambient low salinity waters.

1. Introduction

Analysis of large datasets, originating either from observations such as satellites or model simulations, is an essential component of oceanographic research, a task that entails large memory and computational requirements. The problem is compounded by the variety amongst different datasets in terms of the design of the grid over which it is placed, varying resolutions, and uncertainties in the dataset itself. These characteristics of the dataset impose formidable challenges to the target applications. Analyzing the output of models, in particular, is challenging and time-consuming, especially in the absence of a suitable interactive analysis environment. Here we describe a methodology for the analysis of movement and spreading of the high salinity water that enters the Bay of Bengal (BoB) from the Arabian Sea.

The oceanography of the BoB, especially its low salinity waters exert a dominant control over its temperature distribution and circulation and thus plays a crucial role in breeding monsoon depressions and tropical cyclones (Shenoi et al., 2002). Such weather systems are crucial either for supplying the much-needed water to the hinterland or for the devastation that they bring along their path. Runoff from major river systems such as Ganga, Brahmaputra etc., and heavy monsoon rainfall cause the salinity of the northern part of the BoB to be very low (Behara and Vinayachandran, 2016). In order to maintain the salt balance of the bay, an import of higher salinity water is required. The source of this high salinity water is the Summer Monsoon Current (SMC) which flows eastward from the saltier Arabian Sea into the southern BoB during the summer monsoon (Vinayachandran et al., 1999). Owing to the difference in their densities, the Arabian Sea water flows into the BoB as a sub-surface (between depths of 50 to 150m) high salinity core (HSC) (Vinayachandran et al., 2013). These two sources of contrasting characters create a strong salinity gradient across the BoB, from the region far south to its head near West Bengal. According to Vinayachandran et al. (Vinayachandran et al., 2013), there exists a salt pump in the southern BoB, that episodically draws high salinity water from the HSC and mixes with the relatively lower salinity BoB that is present in the near-surface layers. The pathways of the high salinity water into the rest of the BoB is still not known. Tracking the HSC is one of the key challenges in monsoon oceanography which has hitherto not been accomplished.

The HSC is a continuously evolving mass of water that undergo irregular and unpredictable transformations in its shape as it moves across the BoB, as a function of time. Owing to the interaction between the HSC and lower salinity water that surrounds the HSC, its temperature and salinity undergoes modifications, making it difficult to track HSC in the parameter space. Therefore, if a feature which changes and evolves along with HSC and, at the same time, is robust enough to be tracked over time can be defined, then it is possible that the space and time evolution of the HSC in the BoB can be delineated.

Related work. Automation of water mass tracking using T-S diagrams (Talley et al., 2011) is a popular method in descriptive oceanography. A specific water mass of interest may be identified within the T-S diagram and can be tracked within a TS-space even when multiple water masses mix.

*Corresponding author
upkarsingh@iisc.ac.in (U. Singh); dhipuganesh@gmail.com (T.M. Dhipu); vinay@iisc.ac.in (P.N. Vinayachandran); vijayn@iisc.ac.in (V. Natarajan)

This notion of TS-space was used by Berglund et al. (Berglund et al., 2017) for tracking water masses using a thermohaline stream function. Algorithms for computing these complex thermohaline stream functions do not scale well with increasing data resolution. Additionally, the T-S diagram based methods work only when both temperature and salinity are available for a particular geographical area. This requirement may often be restrictive. Automatic and semi-automatic methods have also been developed for detecting and tracking other structures in the ocean such as upwelling filaments (Nascimento et al., 2012; Artal et al., 2019).

We require the salinity core in the BoB to be clearly defined using suitable geometric and topological structures. We further require the representative structures to be amenable to tracking over time. This is different from previous methods, which do not track the entire HSC. However, similar problems have been studied by the data visualization community. Isosurface extraction and display is a popular approach to visual exploration of a 3D (volumetric) scalar field. An *isosurface* is the preimage of a given scalar value. The isosurface is a collection of points where the scalar field (for example, temperature, salinity, pressure, or speed) maps to a given constant value. A collection of such isosurfaces over a contiguous interval of scalar values constitutes an *isovolume*.

The HSC in the BoB is enclosed by the 35 psu isohaline (Vinayachandran et al., 2013, 2018). Therefore, the representation of the HSC as ≥35 psu isovolume is appropriate and efficient when compared to alternatives such as point clouds and explicit mesh representations. This is particularly true in a dynamic setting where the salinity distribution changes over time. Thus, we focus our study on the extraction and temporal evolution of the ≥35 psu isovolume. Tracking problems have been extensively studied within the field of scalar field visualization (Post et al., 2003; Xie et al., 2019). Solutions to the problem typically begin by defining a geometric feature, say in the form of an isosurface, which is then tracked over time. Bremer et al. (Bremer et al., 2010) compute a track graph using a topological structure called the Reeb graph, which captures the isosurface connectivity, and use it to study temporal evolution of hydrogen flames. Ye et al. (Ye et al., 2015) discuss in-situ methods for feature extraction and tracking based on isosurfaces and depth maps. Widanagamaachchi et al. (Widanagamaachchi et al., 2012) describe an interactive tool based on dynamic tracking graphs to explore large-scale time varying data. Lukasczyk et al. (Lukasczyk et al., 2020) propose dynamic nested tracking graphs, a visual representation of the evolution of isosurfaces and their nested structure. Tzeng et al. (Fan-Yin Tzeng and Kwan-Liu Ma, 2005) apply machine learning techniques towards the extraction of features and design of transfer functions or color maps for visualizing time-varying scalar fields. Doraiswamy et al. (Doraiswamy et al., 2013) introduce a framework to identify and track the movement of cloud systems. Agarwal et al. (Agarwal et al., 2019) describe the use of topological features for identifying interesting events in time-varying multivariate data, whereas Pandey et al. (Pandey et al., 2020) demonstrate the benefit of an integrated geometric and topological approach towards the identification and analysis of Rossby wave packets in the atmosphere. Valsangkar et al. (Valsangkar et al., 2019) introduce a visual exploration framework to identify cyclones, possibly consisting of multiple centres, based on topological features and to visualize their evolution over time. Afzal et al. (Afzal et al., 2019) present RedSeaAtlas, a visual analysis tool for spatio-temporal multivariate data that was created to cater to the needs of scientists who study the Red Sea. Visualization and tracking techniques have played a crucial role in the understanding of various phenomena within oceanography and atmospheric science (Du et al., 2015; Li et al., 2011; Liu et al., 2017; Gad et al., 2018).

Contributions. All analysis frameworks and visualization techniques mentioned above have been designed to cater to requirements and tasks that are specific to the respective application, so they are not directly applicable towards the study of high salinity water in the BoB. It is also notable that many of the above-mentioned methods have a significant user interaction component. A few methods have proposed feature definitions based on well defined structures such as isosurfaces (Bremer et al., 2010; Ye et al., 2015) in data. This approach is not directly applicable because the salinity levels within the HSC is not constant, it varies especially in the outermost layers where it mixes with relatively fresher water. The key challenge in tracking the HSC is that its boundaries are not well defined. Due to the various ocean dynamics processes such as advection by ocean currents, mixing and diffusion, the HSC can be considered to be a continuously evolving mass of salinity that undergoes irregular and unpredictable shape transformations as it moves across BoB. In this paper, we introduce two approaches to represent the HSC and its characteristics, and describe methods to track the movement and evolution of the HSC. The HSC in the southern Bay of Bengal is located between a depth range of 50 – 150m (Vinayachandran et al., 2013), mostly below the intense SMC. The SMC weakens considerably after it crosses the latitude of about 11° N, shedding several eddies on its path (Rath et al., 2019). The path of the HSC into the northern Bay of Bengal largely has not yet been documented which can be mostly attributed to the lack of an appropriate tool. Our primary contributions include

- Introduction of feature definitions of the HSC based on the notion of *fronts* and *skeletons*.
- A parallel algorithm for extracting fronts.
- Algorithms to track front-based and skeleton-based features.
- An interactive visual analysis tool for analyzing HSC propagation in the BoB.
- New documentation on salinity propagation in the BoB.

2. Data

We use reanalysis data from the Nucleus for European Modelling of the Ocean (NEMO) repository (Madec, 2008), at daily resolution for the months June 2016-September 2016, a total of 122 time steps. The data is available in NetCDF format with a latitude-longitude resolution of $1/12°$, faithfully representing the variability of the SMC and associated water masses (Webber et al., 2018). Salinity values are available at 50 vertical levels, which are unequally spaced ranging from 1m apart near the surface to 450m apart near the sea floor, and 22 samples lie within the upper 100m. This data is processed to extract a subset corresponding to the BoB, a geographical region bounded between longitudes $75°E$ and $96°E$, and latitudes $5°S$ to $30°N$. This region is extracted using Climate Data Operators (CDO) command line tools (Schulzweida, 2019), available for manipulating the NetCDF files. The movement of HSC due to currents is observed only in relatively shallow depths. The currents in deeper water are weaker, and the salinity changes within these depths are primarily due to diffusion. Such small scale changes are outside the scope of our study. In this paper, we focus on the study of salinity levels at depth up to 200m.

We use Paraview (Ahrens et al., 2005) for visualizing the salinity field and for interactive exploration of the HSC, its representations, and tracks computed by the methods proposed in this paper. Paraview uses trilinear interpolation within each grid cell to compute a continuous salinity field from the samples on the vertices of a regular grid and uses this field for generating volume visualizations. As mentioned above, the salinity field is sampled at unequally spaced depth levels. We assume linear interpolation along the depth axis and resample the salinity at regular depth levels 1m apart up to 200m, thereby generating the regular grid used for volume visualization. The regularly sampled data is amenable to faster processing and results in improved user interaction and visualization. All computational experiments and visualizations in this paper are based on this data sampled on a regular grid. The geographical map used in the figures is a cropped version of a map from the NASA visible earth catalog (NASA).

3. Front-Based Tracking

The HSC is a continuously evolving water mass. It is difficult to study its movement directly in the parameter space. We propose two approaches, namely the identification and tracking of fronts and skeletons, to study the HSC. These two feature representations provide the necessary abstraction to capture movement and shape, respectively, of the evolving HSC. In this section, we describe the front-based approach to tracking the HSC.

The front is a subset of the boundary of the HSC volume. We describe an efficient parallel algorithm for computing the fronts as connected components of boundary surfaces, an algorithm to track them over time, and a representation of the track as a spatial curve.

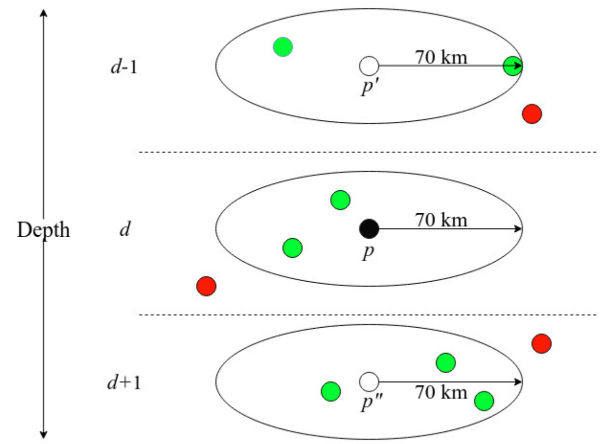


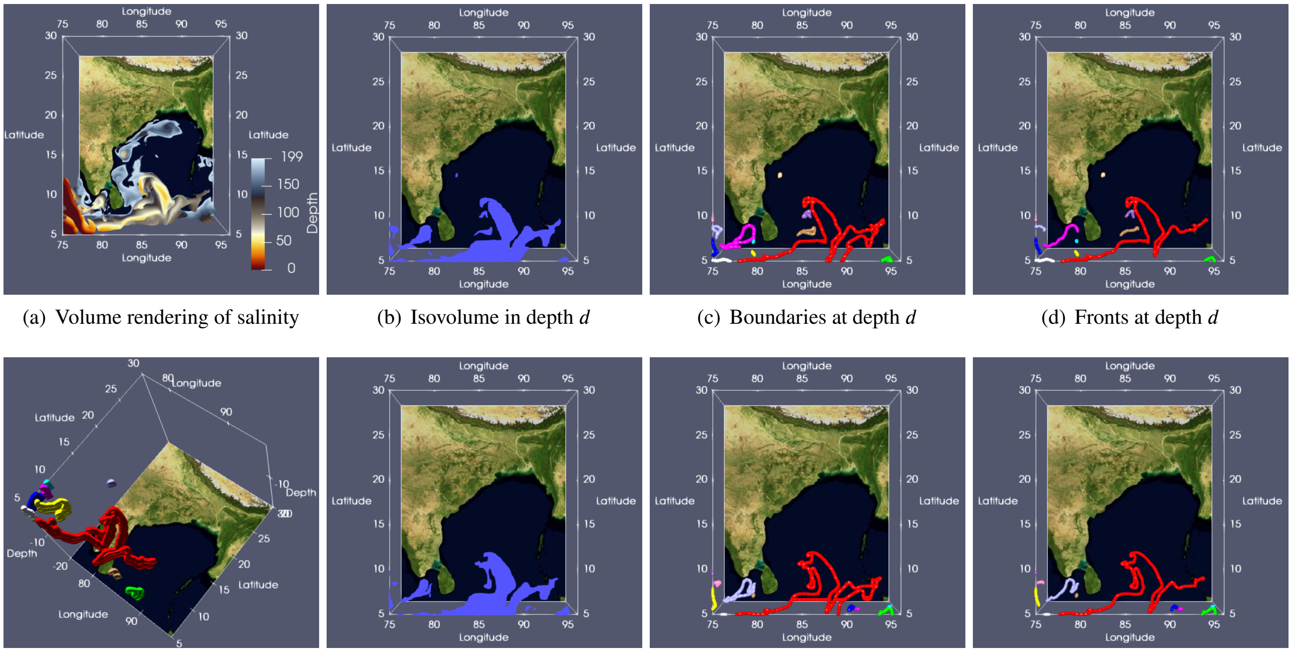
Figure 1: The 2D and 3D neighborhood of a voxel p. The green points represent voxels that lie within a disk of radius 70 km centered at p' (top), p (middle), and p'' (bottom).

Isovolume and surface front. As discussed in Section 2, we interpolate and stored the data on a regular 3D grid. First, we extract the ≥35 psu isovolume and store it as a binary grid where the $1s$ (ones) represent ≥35 psu and $0s$ (zeros) represent other values. This isovolume is a coarse representation of the HSC (Vinayachandran et al., 2013, 2018). We propose the identification and tracking of specific components of the boundary surface of the ≥35 psu isovolume. Specifically, we compute a boundary surface component with a predisposition to move north, declare it as a *front*, and track it over time.

We refer to each sample point in the regular grid as a *voxel*, analogous to a pixel in 2D image or grid. The neighborhood of a voxel is typically defined as the collection of adjacent voxels, 26 in total if we consider adjacency along the grid axes and the diagonals. The neighborhood of the voxel restricted to the horizontal plane consists of 8 voxels. We generalize this notion of neighborhood to incorporate the temporal resolution of the data and the speed of the water current in the BoB. Specifically, data is available to us at a daily resolution and water masses (SMC in BoB) move up to 70 km in a day. We define the 2D neighborhood $N_2(p, d)$ of a voxel p at depth d as the collection of voxels that lie within a disk of radius 70 km. The neighborhood of p in 3D is defined as the union of the 2D neighborhoods of p and the voxels p' and p'' that lie directly above and below p,

$$N_3(p) = N_2(p', d-1) \cup N_2(p, d) \cup N_2(p'', d+1).$$

Figure 1 shows the 2D and 3D neighborhood of a voxel at depth d. We compute the fronts independently within each depth level and stitch them together into a *surface front*. As a first step, we compute the HSC boundary curve within each depth level and segment each connected component of boundary curve into a north facing segment and a south facing segment. Consider the collection of voxels within the ≥35 psu isovolume at time t restricted to a given depth d. This slice of the isovolume is often disconnected, and consists of multiple components. A boundary voxel in an iso-



(a) Volume rendering of salinity (b) Isovolume in depth d (c) Boundaries at depth d (d) Fronts at depth d

(e) Surface fronts (f) Isovolume in depth $d+1$ (g) Boundaries at depth $d+1$ (h) Fronts at depth $d+1$

Figure 2: Isovolume boundary and surface fronts. (a) Salinity distribution over the BoB. (b,f) The ≥35 psu isovolume restricted to depth d and $d+1$. (c,g) Multiple components of the boundary of the isovolume restricted to depth d and $d+1$. (d,h) Fronts extracted from the isovolume boundaries. (e) The surface front computed by stitching the fronts across all depths.

volume slice is adjacent to at least one voxel that lies outside the ≥35 psu isovolume. The collection of boundary voxels constitute the HSC boundary curve $B_{d,t}$ within depth d and time t. As we see in Figures 2(c) and 2(g), a boundary curve may consist of multiple components. The north facing segments of a boundary $B_{d,t}$ is defined as the front $F_{d,t}$. It is computed within each connected component of $B_{d,t}$ by first locating the voxels at the western and eastern extremes and tracing the sequence of boundary voxels between the two as shown in Figures 2(d) and 2(h). We compute the fronts within each depth in parallel.

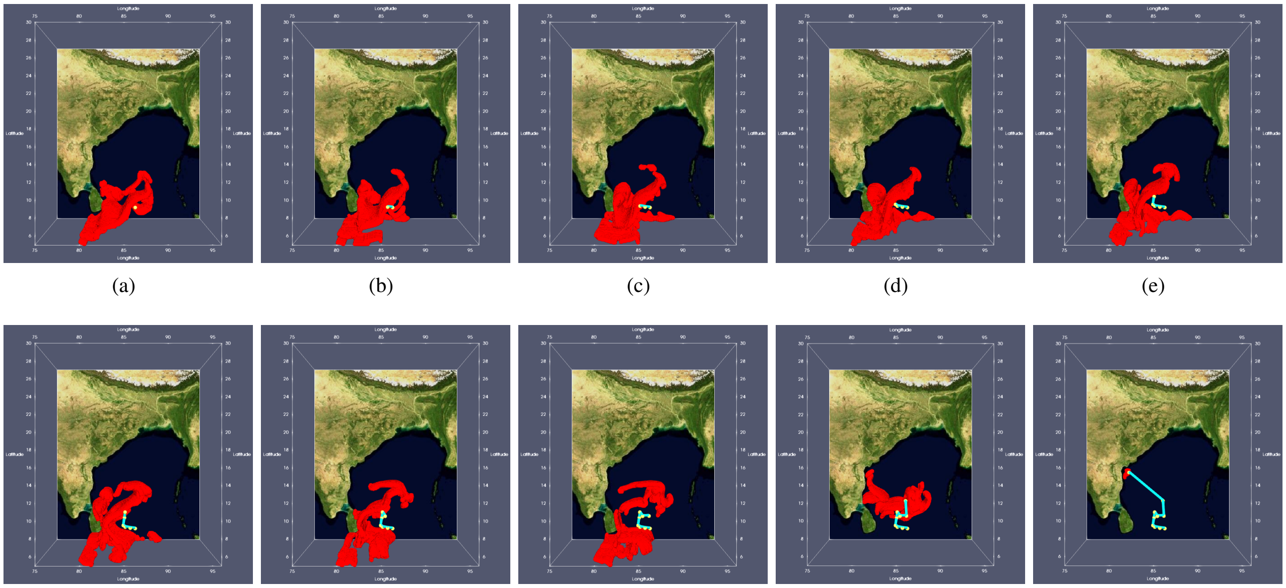
Computing surface fronts. We stitch the fronts computed within each depth using the notion of voxel neighborhood in 3D. Let p be a voxel lying in component C_1 of a front $F_{d,t}$. If a component C_2 of fronts $F_{d,t}$, $F_{d-1,t}$, or $F_{d-1,t}$ contains a voxel q from $N_3(p)$ then we wish to declare that C_1 and C_2 belong to a common surface front.

We construct a graphs whose nodes represent connected components of 2D fronts. There exists an edge between two nodes if the corresponding front components are adjacent to each other as described above. We construct the surface fronts as connected components of this graph. Multiple surface fronts may exist within a time step. We assign a global identifier i to each surface front $SF_{t,i}$ within time step t. Figure 2(e) shows surface fronts from one time step. The surface fronts may be computed efficiently using a parallel connected component algorithm (Han and Wagner, 1990).

Tracking surface fronts. The velocity of SMC in BoB implies that each surface front that we compute can move a maximum of 70 km in one day (one per time step in our data). If the neighborhood $N_3(p)$ of a voxel p in a surface front $SF_{t,i}$ contains a voxel q from $SF_{t+1,j}$, then we declare that $SF_{t+1,i}$ is either a continuation of $SF_{t,i}$ or is created due to a split event at $SF_{t,i}$ or a merge of $SF_{t,i}$ with another surface front. Essentially, we use the voxel neighborhood to identify correspondence between surface fronts and hence track them over time. We construct a track graph $TG_f(V,E)$, where each surface front is a node in V and all continuation/split/merge events are represented as directed arcs from time t to $t+1$. It is easy to deduce that this track graph is a directed acyclic graph (DAG).

We create a visual embedding of the track graph by representing each surface front as a point in space. The point is located at the voxel closest to the centroid of all voxels that belong to the surface front. Arcs of the track graph as displayed as straight line edges between the end point nodes. This embedding serves as a useful visual representation of tracks (paths in the track graph).

Figure 4 shows the track graph computed by our algorithm. Since $TG_f(V,E)$ is a DAG, it contains at least one source and one destination node. The different possible movement of HSC in the BoB is thus captured as the collection of source to destination paths in this track graph. Figure 3 shows the evolution of a single surface front (red) over time together with the representative track (cyan). The number of paths in the track graph is often large in number. We propose two methods in Section 5 for filtering the collection of paths, and hence identify interesting tracks.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 3: (a-j) Evolution of a surface front (red) over time. The sequence of representative voxels (yellow) for the surface front, as it moves in time, together with edges (cyan) connecting the representative voxels constitute a visual representation of the track.

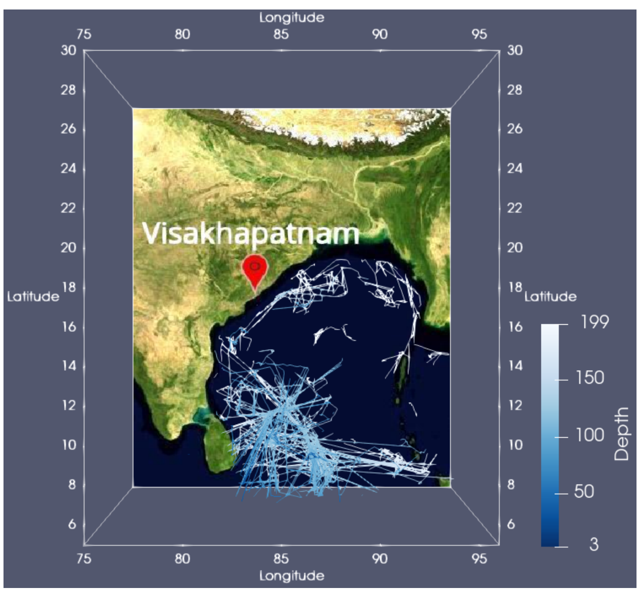


Figure 4: Track graph TG_f computed by our algorithm, which represents movement of the HSC. Paths in the graph represent tracks of individual fronts. These tracks may be studied individually or as clusters to understand, respectively, local and global movement of the HSC. Arcs of the graph are colored based on their depth (blue to white).

4. Skeleton-based Tracking

While front-based tracking captures the movement of the boundary of the HSC, it does not necessarily capture differences between movement of the entire HSC as opposed to an expansion or change of shape of the HSC. We introduce a skeleton-based method that aims to capture the change in shape and track movement of the entire HSC. We describe this method as a four stage pipeline that processes the salinity field sampled over a 3D regular grid:

1. Extract the ≥35 psu isovolume within each time step.
2. Compute a skeleton representation for each isovolume.
3. Compute a track graph TG_s that represents the evolution of the skeletons over time.
4. User-driven selection of skeleton evolution tracks.

Isovolume and skeleton. Topological structures provide abstract representations that are amenable for efficient and robust tracking. Popular methods follow one of two approaches: Morse theory-based or skeleton-based. While Morse theory-based approaches like contour trees and Reeb graphs are powerful and applicable to diverse scenarios, data noise can have an impact on their applicability (Rieck et al., 2017). Hence, we focus on the conceptually simpler skeleton-based approach for tracking the ≥35 psu isovolume.

Topological skeletons describe the connectivity between voxels in a volume represented by a 3D grid. Two voxels that differ in exactly one coordinate by a value of 1 are said to be 6-connected. They are 18-connected if they differ in at most two coordinates, and 26-connected if they differ in one or more of the three coordinates. A volumetric skeleton is defined as an unrooted tree whose nodes consist of such 6/18/26-connected voxels. Volumetric skeletons are derived from 3D volumes for measurement of length, to determine branching and winding structures, and to serve as a feature descriptor for shape matching. Sato et al. (Sato et al., 2000) introduce an efficient skeletonization algorithm to study 3D

CT and MRI scans. Rieck et al. (Rieck et al., 2017) extend this idea to a collection of 2D images to understand the evolution of so-called viscous fingers in fluids. They implement an analysis pipeline to study viscous fingers by tracking them using a sequence of skeletons that change over time. We extend these ideas to the computationally challenging case of 3D domains for tracking the ≥35 psu isovolume.

The first step in the pipeline extracts the ≥35 psu isovolume as described in Section 3.

HSC skeleton. The second step in the pipeline constructs the skeletal structure of isovolumes computed for each time step. The TEASAR (Sato et al., 2000) algorithm employs a conceptually simple approach for skeletonizing the isovolume. We use this algorithm to extract the skeletal structure as a collection of paths in a graph that represents the isovolume. Hereafter, we use the term *HSC skeleton* to refer to skeletons extracted from the HSC.

The algorithm starts by finding a root voxel in a 3D volume, then launches Dijkstra's shortest path algorithm through a penalty field in the isovolume to reach the most distant unvisited voxel. Voxels in the isovolume are considered as nodes of the isovolume graph and the edge set is given by pairs of 26-connected voxels. After each pass through the isovolume, an iterative thinning procedure marks voxels in the neighborhood of the path as visited and removes them from the graph for future passes. The thinning is performed using a sphere, which determines the size of the neighborhood of voxels in the path that are to be marked visited. Thinning ensures that paths computed in two different passes are disconnected. The algorithm uses a few parameters:

- a scaling term s_α and a constant s_c. These parameters control the radius of the sphere

$$r(x, y, z) = s_\alpha \cdot d_B(x, y, z) + s_c,$$

  where $d_B(x, y, z)$ represents the distance of a voxel with coordinates (x, y, z) from the boundary of the isovolume.

- a size threshold s_n. This threshold is used to cull connected components of the skeleton. Components with fewer than s_n nodes are considered as *dust pieces* and discarded.

After the skeletonization step, we obtain a sequence of HSC skeletons S_i, one corresponding to each time step t_i. Figure 5 shows the HSC skeleton extracted from timesteps 33 and 34.

Skeleton tracking. The third step in the pipeline tracks the skeletons that were constructed in the previous step across successive time steps and creates a representation of their evolution. Rieck et al. (Rieck et al., 2017) discuss a tracking algorithm for tracking skeletons of viscous fingers in 2D. We extend this algorithm to track HSC skeletons in 3D. This extension is made non-trivial due to the size of the isovolume and additional requirement of computing tracks that represent significant movement of the skeleton. Our algorithm begins by partitioning the connected components (paths) of each skeleton into small line segments. We compute *segment-to-segment* correspondences instead of *voxel-to-voxel* correspondences, which ensures a good approximation while ensuring a good computational speedup.

The HSC skeleton is a collection of directed paths. For uniformity, we represent each line segment using the start end point. Given two time steps t_i and t_{i+1}, we assign every segment s in a skeleton S_i to the segment s' in skeleton S_{i+1} that satisfies

$$s' = \underset{r \in S_{i+1}}{\arg\min}\ dist(s, r),$$

where $dist(,)$ is the Euclidean distance. Likewise, we assign every segment in S_{i+1} to its nearest neighbor in S_i. This assignment results in a set of directed matches between S_i and S_{i+1}. Each skeleton segment is guaranteed to occur at least once in the above set of correspondences. We refer to matches from S_i to S_{i+1} as *forward matches*, and matches in the other direction as *backward matches*. The collection of forward and backward matches across all time steps is stored as a track graph TG_s. Figure 6 shows the forward matches between skeletons S_{33} and S_{34} and the track graph that represents movements of all segments of HSC skeletons.

Query-driven track computation. The final stage in the skeleton-based tracking pipeline is user-driven identification and display of skeleton movement. The motivation for including this final stage is to generate representative tracks that describes the overall movement of the HSC with a focus on regions selected by the oceanographer. The user interactively selects a collection of voxels of interest in the HSC skeleton after observing the track graph TG_s or individual skeletons. Tracks passing through the selected voxels are computed via forward and backward path tracing and displayed as an indicator of propagation routes of salinity. Figure 9 shows tracks computed using this method that depict the movement of the HSC skeleton towards Visakhapatnam.

5. Visual Analysis Tool Design

All algorithms described in this paper are implemented in Python and we use Paraview (Ahrens et al., 2005) for visualizing the salinity field, HSC, and tracks. Multiple python scripts execute either independently or within the Paraview framework to compute the results. In this section, we describe the functionality and usage of the python scripts and hence the design of the visual analysis tool. A comprehensive user manual together with a discussion of dependencies will be made available in the software documentation.

Input parameters. All parameters used by the algorithm and display routines are specified in a text file. The parameters are assigned default values that work for most experiments, but may be tuned by the user. For example, the latitude-longitude resolution is set to a default value of $1/12°$. By default, the data is resampled on a regular grid as discussed in Section 2. However, this may be updated to skip

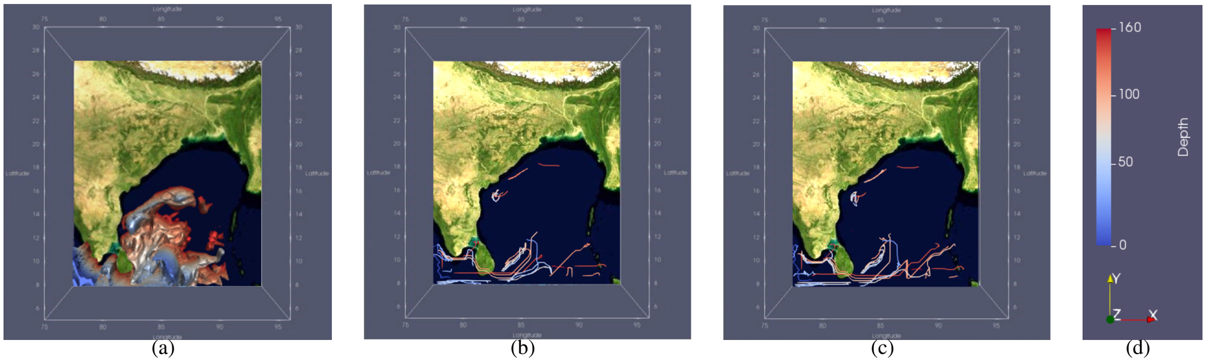


Figure 5: (a) The ≥35 psu isovolume. (b,c) HSC skeletons for timesteps 33 and 34, respectively. (d) The depth in the isovolume and skeleton figures is mapped to color using a blue-red colormap.

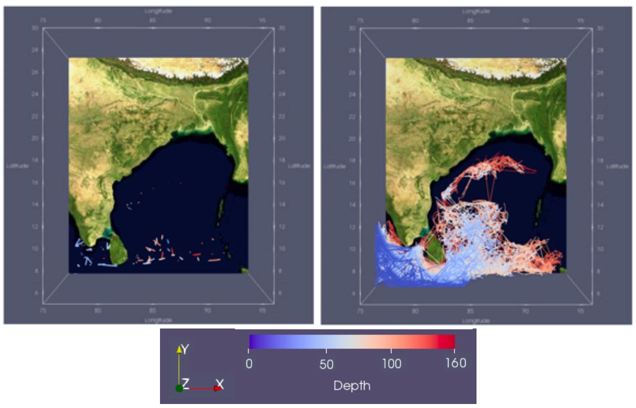


Figure 6: (left) Forward matches between HSC skeletons S_{33} and S_{34}. (right) All forward and backward matches are collected together into a track graph TG_s that can be queried to extract interesting tracks or to identify tracks within a specific region or time period of interest.

the resampling if the input is already available on a regular grid in netCDF format. Similarly, all display parameters and options may be stored within a Paraview state file to quickly load previously generated visualizations.

Front tracking. For visual analysis of the evolution of surface fronts, we extract 50 (a user-defined constant) paths in the track graph TG_f. Prior to this, we establish a focus region, namely the BoB, by filtering TG_f to remove surface fronts that lie west of Sri Lanka (using a longitude threshold). We cluster the paths depending on their source and destination points. A standalone python script `TrackGraph.py` computes the HSC boundary, surface fronts, and TG_f. It stores them in a VTP file that can be read by Paraview for visualization. A second script `LongPaths.py` enumerates paths in TG_f (typically directed north), selects the longest k paths, clusters them into bins specified by their end point locations, and finally applies geometric simplification to generate straight polylines that represent the tracks. It stores the paths in NPY files for further analysis as required. The output of these script depend on user specified parameters such as number of paths, number of clusters of paths, region of interest specified as latitude-longitude thresholds.

We use Paraview to display the track graph and the paths using meaningful colormaps, see Figures 4 and 7. The system supports queries, where the user may select an area using Paraview which translates to a collection of nodes from the track graph. A script `SelectPaths.py` works as a programmable filter within Paraview to support the query. It computes all paths in TG_f passing through the selected nodes and displays the corresponding tracks. Optionally, the user can load a predefined state file to interact with a default visualization within Paraview consisting of the isovolume and a collection of tracks.

Skeleton tracking. Similar to front tracking, the visual analysis pipeline for skeleton tracking is designed such that it begins with an automatic mode that constructs the skeletons and the track graph TG_s followed by an interactive mode where the user selects points of interest to display select tracks and hence explore the movement of the HSC. The automatic mode involves utilizing two major capabilities of Parview – to programmatically invoke filter functions using the pvpython utility and to run external scripts along with pvpython. A script `skeletonize.py` implements the TEASAR algorithm (Sato et al., 2000) to construct the skeleton of isovolumes at all time steps. Each skeleton is stored as a NetworkX graph. A second script `tracking.py` traverses the NetworkX graph to compute TG_s and stores it in VTP format that can be readily processed by Paraview for visualization and exploration.

The oceanographer interacts with the set of temporal tracks stored in TG_s using a collection of built-in and programmable filters that help select points of interest. Beginning with the user selected points and tracking backwards and forwards in time, the system identifies tracks that represent likely propagation routes of the salinity core.

6. Results

We now discuss our observations on salinity movement and the evolution of the HSC in the BoB using the methods described in the previous sections.

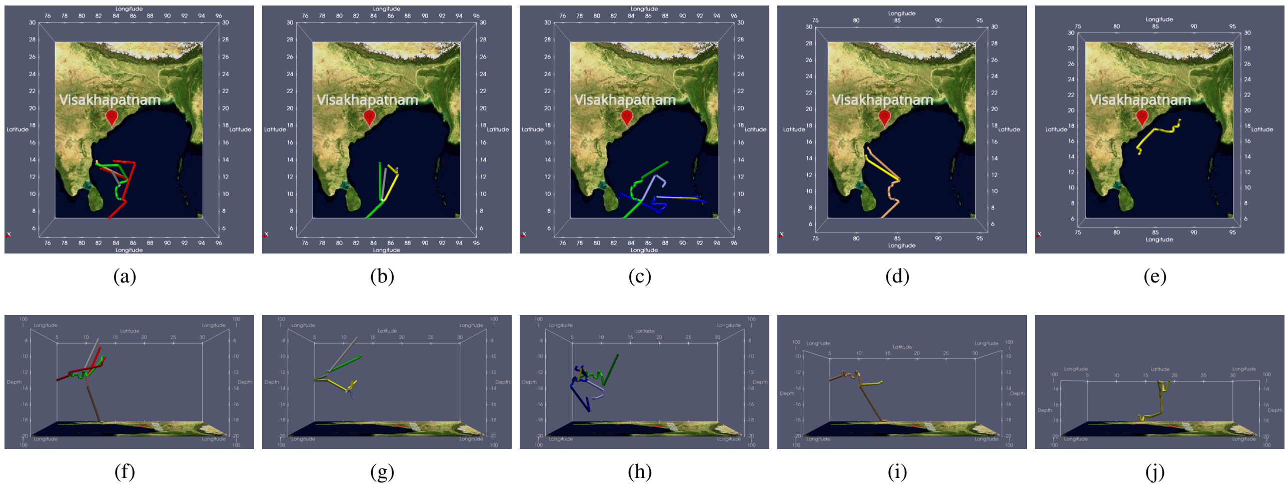


Figure 7: Different tracks extracted from the track graph TG_f, grouped together based on their source and destination node in the graph. (a,b,c,d,e) top view. (f,g,h,i,j) corresponding side view from east.

HSC Movement. The HSC, after entering the BoB, undergoes considerable transformation in its course. The energetic fluctuations of the SMC, its meandering and eddy shedding behaviors (Rath et al., 2019) induce space time variations on depth of penetration, intensity, and advancement of HSC into the northern BoB. The main branch of the SMC splits into multiple branches (Webber et al., 2018; George et al., 2019) and consequently, a smooth northward or northeastward flow of the HSC is not apparent. The video `hsc-isovolume` accompanying this paper shows the movement of the HSC during June-September, 2016. The HSC is represented as an isovolume and visualized using volume rendering, which maps depth to color and transparency. Among the distinct patterns, the following appears to be prominent: (1) Towards the peak of the summer monsoon season, an anticyclonic (clockwise) eddy forms which recirculate the HSC back to its core as the core itself propagates westwards. The core eventually collapses by the end of the season after it encounters the coast of Sri Lanka. (2) One branch of the HSC travels eastward initially but, later in the season, this patch moves westward to merge with the main axis of the SMC owing to the influence of the westward propagating Rossby waves that dominate the dynamics in this regions (Vinayachandran and Yamagata, 1998; Webber et al., 2018; Rath et al., 2019). (3) Isolated patches of high salinity water can be seen around Sri Lanka at deeper depths (150 – 200m) throughout the season. (4) The leading edge of the HSC bends anticlockwise, meanders, detaches from the main axis as an eddy, and then moves northwestward towards Visakhapatnam retaining its high salinity character for a long distance, see also Figures 7 and 9.

Validation. The accompanying video `hsc-front-track` shows how the surface front of the isovolume is correctly tracked and represented as a path in 3D. The isovolumes tracked by the method presented in this study are consistent with previously published data. Vinayachandran et al. (Vinayachandran et al., 2013) clearly identify the region in the southern BoB where the HSC enters from the Arabian Sea, after surrounding the Sri Lanka Dome(SLD) (Sanchez-Franks et al., 2019). The westward propagation of SMC and its separation from the east coast of Sri Lanka is consistent with the BoBBLE observations presented by Vinayachandran et al. (Vinayachandran et al., 2018) and Webber et al. (Webber et al., 2018). George et al. (George et al., 2019) have delineated the forking of the HSC into three directions after entering the BoB and turning around the SLD. Our methods reproduce visualizations of these observations (Figure 9). The eastward tracks of HSC and its re-circulation are consistent with the analysis presented by George et al. (George et al., 2019). Detailed documentation of the flow of the HSC into the northern BoB is not yet available and we hope that the snapshots presented here (Figures 7(e,j) and 8) will provide useful guidelines for future efforts in this direction.

Visual analysis of tracks. Several new features of the paths of the HSC in the northern Bay of Bengal have emerged from this study. The model outputs used here analyzed products that ingest available observations into it, which improve the reliability of the results. The longest track of HSC, after entering BoB, passes through the centre and then turns west towards the coast at Visakhapatnam. Such paths can be seen in Figure 7(d) which was generated using front-based tracking method and Figure 9 which was generated using the skeleton-based tracking method. Both figures show similar trends in the temporal evolution of the HSC. There have been observations of high salinity patches in the depth range of 100 – 200 m (Sasamal, 1990), which are in support of the results obtained from our analysis.

Some tracks start near Sri Lanka and terminate near the Visakhapatnam coast (Figure 7(a,f,d,i)) while passing through the BoB, others stop near the centre of BoB (Figure 7(b,g), and a few turn toward the coast of Andaman and Nicobar

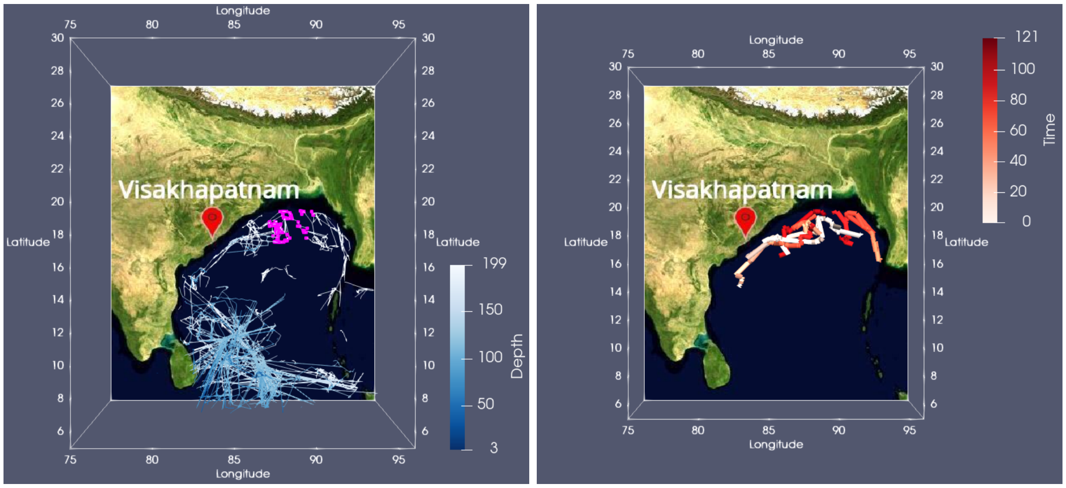


Figure 8: Interactive queries to study local movement of HSC. (left) The track graph TG_f colored (blue to white) based on depth and a collection of graph nodes (pink) selected by the user. The selected nodes correspond to representative voxels of different surface fronts. (right) Tracks that contain the selected nodes are extracted from the graph and displayed on demand.

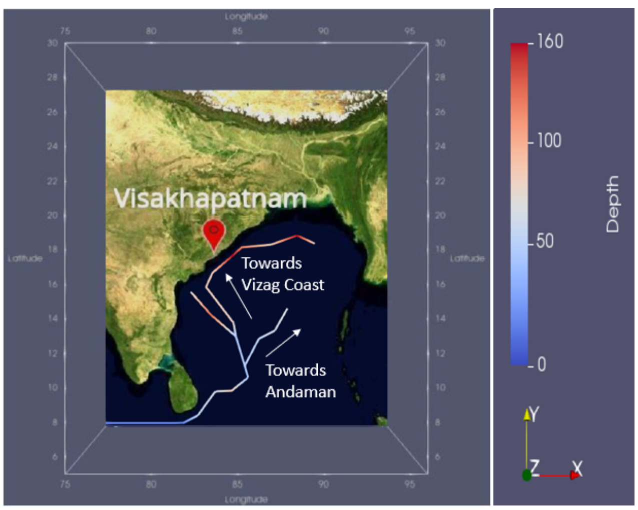


Figure 9: Representative tracks that depict movement of the HSC skeleton. Tracks are computed against a user specified query consisting of points near south of BoB, Visakhapatnam coast, and Andaman coast.

islands (Figure 7(c,h). We observe a similar behavior even when we increase the number of extracted tracks from 50 to 100. This offers an alternate method to delineate the path of the SMC and the HSC. After reaching almost the centre of BoB it forks and moves in two directions. Tracks computed using the skeleton-based approach corroborates this finding, see Figure 9. Vinayachandran et al. (Vinayachandran et al., 2013) report about the SLD, which is a cyclonic (anticlockwise) eddy caused by cyclonic curl in the local wind field. The SLD is evident in our results (Figure 9). We observed a considerable reduction in the number of matches (arcs) in TG_s while passing through this region.

We also observed a movement of high salinity water from Visakhapatnam coast towards north, along the coast of India. From the track graph, it is clear that this track is not a continuation of the HSC component that reached Visakhapatnam coast because this path originates at a time when the HSC is still entering the BoB near Sri Lanka. Using the query system, we can extract the required tracks (Figures 8 and 9). This movement, most probably, is a result of high salinity water pockets near Visakhapatnam coast persisting from the previous year.

Computational performance. All steps in the track graph computation require a linear running time because we iterate over the regular grid, boundary voxels, and surface fronts only a constant number of times. Hence, the worst case runtime complexity of the track graph computation is linear in the input grid size. Computing one path in the track graph takes $O(|E| \log |V|)$ time using Dijkstra's algorithm (Cormen et al., 2009, Chapter 24). But, the time required for computing multiple paths is a function of both the number of paths and the size of the track graph. The worst case running time for computing the skeletons also requires repeated application of Dijkstra's algorithm and takes $O(|E| \log |V|)$ time.

7. Conclusions

The skeleton-based tracking method provides an overall picture of how the shape of HSC evolves over time in the BoB because the skeleton represents the global shape of the isovolume. The front-based tracking method provides concrete tracks describing the movement of the HSC. These complementary methods produce results that corroborate each other. We validate the results by comparing it with prior observations and via visual comparison of the tracks and volume rendering of the HSC, see videos in supplementary material. In future, we wish to incorporate a model of salt diffusion across the fronts to further understand the tracks and salinity distribution within the BoB. We will also incorporate other physical quantities measured over the region towards an improved understanding of the interplay between monsoons and oceanography of the region.

The tracking methods discussed in this paper is suitable for application to other regions as well for tracking water masses. For example, these methods can be used for tracking the flow of Mediterranean Sea Water in the Atlantic Ocean (Richardson et al., 2000) or to study the movement of North Atlantic Deep Water (Dickson and Brown, 1994) on a global scale. The method is, in principle, applicable to the study of propagation of salinity or other physical quantities irrespective of the region. However, the parameters may have to be tuned appropriately. The track graph is computed within each depth slice independently. This allows the method to be applicable to larger regions. Subsequent track search and queries do require the entire track graph to be resident in memory, and hence require improvements in order to scale to large data sizes. Parallelizing different steps of the method will lead to better run-time performance.

Acknowledgements

This study was funded by an IoE grant from IISc Bangalore. US is supported by a scholarship from MHRD, Govt. of India. PNV acknowledges partial support from the BoBBLE (Bay of Bengal Boundary Layer Experiment) project funded by the Ministry of Earth Sciences, Govt. of India and the J. C. Bose Fellowship awarded by the SERB, DST, Govt. of India. VN is supported by a Swarnajayanti Fellowship from the Department of Science and Technology, India (DST/SJF/ETA-02/2015-16) and a Mindtree Chair research grant.

Computer Code Availability

The codes and scripts for this research are available at https://bitbucket.org/vgl_iisc/bob-salinity-visualization.

CRediT authorship contribution statement

Upkar Singh: Methodology - front computation and visual analysis, Investigation, Visualization, Software, Writing - Original Draft. **T.M. Dhipu:** Methodology - skeleton computation, Investigation, Software, Writing - Original Draft. **P. N. Vinayachandran:** Conceptualization, Writing - Review and Editing. **Vijay Natarajan:** Conceptualization of this study, Methodology, Writing - Review and Editing.

References

Afzal, S., Ghani, S., Tissington, G., Langodan, S., Dasari, H.P., Raitsos, D., Gittings, J., Jamil, T., Srinivasan, M., Hoteit, I., 2019. RedSeaAtlas: A visual analytics tool for spatio-temporal multivariate data of the Red sea, in: Workshop on Visualization in Environmental Sciences (EnvirVis), pp. 25–32. doi:10.2312/envirvis.20191101.

Agarwal, T., Chattopadhyay, A., Natarajan, V., 2019. Topological feature search in time-varying multifield data. CoRR abs/1911.00687. URL: http://arxiv.org/abs/1911.00687, arXiv:1911.00687.

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

Artal, O., Sepúlveda, H.H., Mery, D., Pieringer, C., 2019. Detecting and characterizing upwelling filaments in a numerical ocean model. Computers & Geosciences 122, 25–34.

Behara, A., Vinayachandran, P., 2016. An ogcm study of the impact of rain and river water forcing on the bay of bengal. Journal of Geophysical Research: Oceans 121, 2425–2446.

Berglund, S., Döös, K., Nycander, J., 2017. Lagrangian tracing of the water–mass transformations in the atlantic ocean. Tellus A: Dynamic Meteorology and Oceanography 69, 1306311. doi:10.1080/16000870.2017.1306311.

Bremer, P., Weber, G., Pascucci, V., Day, M., Bell, J., 2010. Analyzing and tracking burning structures in lean premixed hydrogen flames. IEEE Transactions on Visualization and Computer Graphics 16, 248–260. doi:10.1109/TVCG.2009.69.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to Algorithms, Third Edition. 3rd ed., The MIT Press.

Dickson, R.R., Brown, J., 1994. The production of north atlantic deep water: Sources, rates, and pathways. Journal of Geophysical Research: Oceans 99, 12319–12341. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/94JC00530, doi:https://doi.org/10.1029/94JC00530, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/94JC00530.

Doraiswamy, H., Natarajan, V., Nanjundiah, R.S., 2013. An exploration framework to identify and track movement of cloud systems. IEEE Transactions on Visualization and Computer Graphics 19, 2896–2905. doi:10.1109/TVCG.2013.131.

Du, Z., Fang, L., Bai, Y., Zhang, F., Liu, R., 2015. Spatio-temporal visualization of air–sea co2 flux and carbon budget using volume rendering. Computers & Geosciences 77, 77–86.

Fan-Yin Tzeng, Kwan-Liu Ma, 2005. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations, in: SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, pp. 6–6. doi:10.1109/SC.2005.37.

Gad, M.A., Elshehaly, M.H., Gračanin, D., Elmongui, H.G., 2018. A tracking analyst for large 3d spatiotemporal data from multiple sources (case study: Tracking volcanic eruptions in the atmosphere). Computers & Geosciences 111, 283–293.

George, J.V., Vinayachandran, P.N., Vijith, V., Thushara, V., Nayak, A.A., Pargaonkar, S.M., Amol, P., Vijaykumar, K., Matthews, A.J., 2019. Mechanisms of barrier layer formation and erosion from in situ observations in the bay of bengal. Journal of Physical Oceanography 49, 1183 – 1200. URL: https://journals.ametsoc.org/view/journals/phoc/49/5/jpo-d-18-0204.1.xml, doi:10.1175/JPO-D-18-0204.1.

Han, Y., Wagner, R.A., 1990. An efficient and fast parallel-connected component algorithm. J. ACM 37, 626–642. URL: https://doi.org/10.1145/79147.214077, doi:10.1145/79147.214077.

Li, W., Chen, G., Kong, Q., Wang, Z., Qian, C., 2011. A vr-ocean system for interactive geospatial analysis and 4d visualization of the marine environment around antarctica. Computers & Geosciences 37, 1743–1751.

Liu, S., Chen, G., Yao, S., Tian, F., Liu, W., 2017. A framework for interactive visual analysis of heterogeneous marine data in an integrated problem solving environment. Computers & Geosciences 104, 20–28.

Lukasczyk, J., Garth, C., Weber, G.H., Biedert, T., Maciejewski, R., Leitte, H., 2020. Dynamic nested tracking graphs. IEEE Transactions on Visualization and Computer Graphics 26, 249–258. doi:10.1109/TVCG.2019.2934368.

Madec, G., 2008. NEMO ocean engine. Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27, ISSN No 1288-1619.

NASA, . Visible earth : A catalog of nasa images and animations of our home planet. URL: https://visibleearth.nasa.gov/collection/1484/blue-marble.

Nascimento, S., Franco, P., Sousa, F., Dias, J., Neves, F., 2012. Automated computational delimitation of sst upwelling areas using fuzzy clustering. Computers & Geosciences 43, 207–216.

Pandey, K., Monteiro, J.M., Natarajan, V., 2020. An integrated geometric and topological approach for the identification and visual analysis of rossby wave packets. Monthly Weather Review 148, 3139 – 3155. URL: https://journals.ametsoc.org/view/journals/mwre/148/8/mwrD200014.xml, doi:10.1175/MWR-D-20-0014.1.

Post, F.H., Vrolijk, B., Hauser, H., Laramee, R.S., Doleisch, H., 2003. The state of the art in flow visualisation: Feature extraction and tracking. Computer Graphics Forum 22, 775–792.

Rath, S., Vinayachandran, P., Behara, A., Neema, C., 2019. Dynamics of summer monsoon current around Sri Lanka. Ocean Dynamics 69, 1133–1154.

Richardson, P., Bower, A., Zenk, W., 2000. A census of meddies tracked by floats. Progress in Oceanography 45, 209–250. URL: https://www.sciencedirect.com/science/article/pii/S0079661199000531, doi:https://doi.org/10.1016/S0079-6611(99)00053-1.

Rieck, B., Sadlo, F., Leitte, H., 2017. Persistence concepts for 2d skeleton evolution analysis, in: Topological Methods in Data Analysis and Visualization, Springer. pp. 139–154.

Sanchez-Franks, A., Webber, B.G.M., King, B.A., Vinayachandran, P.N., Matthews, A.J., Sheehan, P.M.F., Behara, A., Neema, C.P., 2019. The railroad switch effect of seasonally reversing currents on the bay of bengal high-salinity core. Geophysical Research Letters 46, 6005–6014. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL082208, doi:https://doi.org/10.1029/2019GL082208.

Sasamal, S., 1990. High saline waters in bay of bengal. Proc. Indian Academy of Sciences-Earth and Planetary Sciences 99, 367–381.

Sato, M., Bitter, I., Bender, M.A., Kaufman, A.E., Nakajima, M., 2000. TEASAR: Tree-structure extraction algorithm for accurate and robust skeletons, in: Proc. Pacific Conf. Computer Graphics and Applications, pp. 281–449.

Schulzweida, U., 2019. Cdo user guide. URL: https://doi.org/10.5281/zenodo.3539275, doi:10.5281/zenodo.3539275.

Shenoi, S.S.C., Shankar, D., Shetye, S.R., 2002. Differences in heat budgets of the near-surface arabian sea and bay of bengal: Implications for the summer monsoon. Journal of Geophysical Research: Oceans 107, 5–1–5–14. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2000JC000679, doi:https://doi.org/10.1029/2000JC000679.

Talley, L., Pickard, G., Emery, W., Swift, J., 2011. Descriptive physical oceanography: An introduction: Sixth edition. Descriptive Physical Oceanography: An Introduction: Sixth Edition , 1–555.

Valsangkar, A.A., Monteiro, J.M., Narayanan, V., Hotz, I., Natarajan, V., 2019. An exploratory framework for cyclone identification and tracking. IEEE Transactions on Visualization and Computer Graphics 25, 1460–1473. doi:10.1109/TVCG.2018.2810068.

Vinayachandran, P., Masumoto, Y., Mikawa, T., Yamagata, T., 1999. Intrusion of the southwest monsoon current into the bay of bengal. Journal of Geophysical Research: Oceans 104, 11077–11085.

Vinayachandran, P., Matthews, A.J., Kumar, K.V., Sanchez-Franks, A., Thushara, V., George, J., Vijith, V., Webber, B.G., Queste, B.Y., Roy, R., et al., 2018. BoBBLE: Ocean–atmosphere interaction and its impact on the South Asian monsoon. Bulletin of the American Meteorological Society 99, 1569–1587.

Vinayachandran, P., Yamagata, T., 1998. Monsoon response of the sea around Sri Lanka: generation of thermal domes and anticyclonic vortices. Journal of Physical Oceanography 28, 1946–1960.

Vinayachandran, P.N., Shankar, D., Vernekar, S., Sandeep, K.K., Amol, P., Neema, C.P., Chatterjee, A., 2013. A summer monsoon pump to keep the bay of bengal salty. Geophysical Research Letters 40, 1777–1782. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/grl.50274, doi:https://doi.org/10.1002/grl.50274.

Webber, B.G., Matthews, A.J., Vinayachandran, P., Neema, C., Sanchez-Franks, A., Vijith, V., Amol, P., Baranowski, D.B., 2018. The dynamics of the southwest monsoon current in 2016 from high-resolution in situ observations and models. Journal of Physical Oceanography 48, 2259–2282.

Widanagamaachchi, W., Christensen, C., Pascucci, V., Bremer, P., 2012. Interactive exploration of large-scale time-varying data using dynamic tracking graphs, in: IEEE Symp. Large Data Analysis and Visualization (LDAV), pp. 9–17. doi:10.1109/LDAV.2012.6378962.

Xie, C., Li, M., Wang, H., Dong, J., 2019. A survey on visual analysis of ocean data. Visual Informatics 3, 113–128.

Ye, Y.C., Wang, Y., Miller, R., Ma, K., Ono, K., 2015. In situ depth maps based feature extraction and tracking, in: IEEE Symp. Large Data Analysis and Visualization (LDAV), pp. 1–8. doi:10.1109/LDAV.2015.7348065.