Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Multiple Hypothesis Object Tracking For Unsupervised Self-Learning: An Ocean Eddy Tracking Application

James H. Faghmous, Muhammed Uluyol, Luke Styles, Matthew Le,

Varun Mithal, Shyam Boriah and **Vipin Kumar**
Department of Computer Science and Engineering
University of Minnesota

Abstract

Mesoscale ocean eddies transport heat, salt, energy, and nutrients across oceans. As a result, accurately identifying and tracking such phenomena are crucial for understanding ocean dynamics and marine ecosystem sustainability. Traditionally, ocean eddies are monitored through two phases: identification and tracking. A major challenge for such an approach is that the tracking phase is dependent on the performance of the identification scheme, which can be susceptible to noise and sampling errors. In this paper, we focus on tracking, and introduce the concept of multiple hypothesis assignment (MHA), which extends traditional multiple hypothesis tracking for cases where the features tracked are noisy or uncertain. Under this scheme, features are assigned to multiple potential tracks, and the final assignment is deferred until more data are available to make a relatively unambiguous decision. Unlike the most widely used methods in the eddy tracking literature, MHA uses contextual spatio-temporal information to take corrective measures autonomously on the detection step *a posteriori* and performs significantly better in the presence of noise. This study is also the first to empirically analyze the relative robustness of eddy tracking algorithms.

1 Introduction

Understanding how the oceans and their ecosystems will respond to global changes in temperature and atmospheric CO_2 levels remains a topic of intense scientific and societal interest (Caldeira, Wickett, and others 2003; Hoegh-Guldberg and Bruno 2010). In order to project a future response accurately, it is crucial that current and past ocean dynamics be well understood. Mesoscale ocean eddies (hereafter eddies) are coherent rotating structures of ocean spanning tens to hundreds of kilometers and lasting a few days to several months. Eddies are critical phenomena as they dominate the ocean's kinetic energy and are responsible for the transport of heat, salt, nutrients, and energy across the oceans (Fu et al. 2010).

Given their importance to ocean dynamics, identifying and tracking eddies has been an active field of research (see (Chelton, Schlax, and Samelson 2011) for a review). Traditionally, autonomous eddy monitoring is performed in two independent steps. First, eddy-like features are identified in successive frames of satellite data. Second, the eddy-like features are tracked across time by associating each feature in one frame to another feature in the following time-step. The focus of this paper is on the tracking phase of this two-step process.

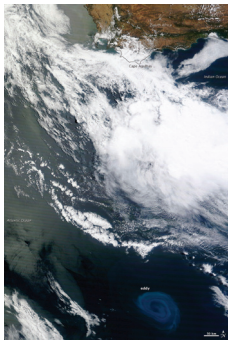

Figure 1: An eddy near the southern tip of the African continental mass (near bottom right corner). This large eddy (150km in diameter) moved heat and nutrients from the warm Indian Ocean to the cooler Atlantic Ocean. Figure courtesy of NASA.

Recently, Chelton, Schlax, and Samelson (2011), hereby CH11, performed the most comprehensive study in autonomous global eddy identification and tracking in sea surface height (SSH) altimeter data. These results were subsequently used in a groundbreaking study published in the journal *Science*, that empirically detailed the impact eddies have on marine biological systems (Chelton et al. 2011). However, tracking eddies with the most widely used eddy-tracking technique has two major limitations: first it is inexorably dependent on the performance of the automatic eddy identification algorithms. Such methods are highly susceptible to noise and have been known to "miss" features for a few time steps and merge features in close proximity (Chelton, Schlax, and Samelson 2011). Second, most eddy tracking methods employ a local nearest neighbor (LNN) tracking algorithm that provides little flexibility for previous errors in detection.

At its core, eddy tracking is a *data association* problem and lends itself to a family of multi-target tracking applications (Cham and Rehg 1999; Han et al. 2004), most notably deferred-logic (Poore 1995) techniques such as multiple hy-

pothesis tracking (Reid 1979) that defer making uncertain associations until more data are available. Such methods have been especially useful in cluttered or noisy environments (Bar-Shalom and Li 1995). Extensive work has been done in multi-target tracking and its applications to security (Blackman 2004; Oh, Russell, and Sastry 2009), computer vision (Cox and Hingorani 1996), and autonomous agents (Nieto et al. 2003; Montemerlo et al. 2003). However, the most general multiple hypothesis tracking (MHT) methods do not address the challenges of tracking noisy (merged eddies in high density areas) and incomplete data (eddies disappearing for a few timesteps without actually dying). As a result, we introduce the notion of *multiple hypothesis assignment* to extend MHT for physical processes applications. MHA is inspired by MHT in terms of (i) tracking multiple objects over several time-steps; and (ii) deferred logic by maintaining multiple plausible assignments until a relatively more certain assignment is possible. We augment the basic MHT approach to address the nature of climate data by introducing a *lookahead* (incompleteness of input) and *unsupervised self-learning* (noise within input features) capability. The unsupervised learning feature is unique in the literature as we use it to autonomously identify errors in the eddy identification phase and subsequently update the eddy-feature database with the correct features by removing the artificially large feature and adding several (corrected) smaller features.

Our method provides several improvements over the state-of-the-art: First, it is more resilient to noisy observations which is a major concern in satellite products. Second, our tracking method is able to maintain tracks even in the event of eddies "disappearing" for a few time-steps and then reappearing due to noise and other sampling constraints. Finally, we employ heuristics to identify and correct possible errors in the eddy identification step in an unsupervised fashion. We show with a variety of empirical and qualitative experiments that MHA's improvements have a significant impact on the resulting eddy tracks compared to the most widely used method in the literature.

2 Background and Problem Formulation

Traditionally, the automatic detection and tracking of ocean eddies were achieved using sea surface temperature or ocean color satellite data (Pegau, Boss, and Martínez 2002; Fernandes 2008; Dong et al. 2011). The advent of SSH observations from satellite radar altimeters provided researchers with an unprecedented opportunity to study eddy dynamics on a global scale. This is because eddy behavior is intimately linked to SSH. Eddies are generally classified by their rotational direction. Cyclonic eddies rotate counter-clockwise (in the Northern Hemisphere), while anti-cyclonic eddies rotate clockwise. As a result, cyclonic eddies cause a decrease in SSH, while anti-cyclonic eddies cause an increase in SSH. Such impact allows us to identify ocean eddies in SSH satellite data, where cyclonic eddies manifest as closed contoured negative SSH anomalies and anti-cyclonic eddies as positive SSH anomalies.

Eddies are identified in the SSH field by assigning binary values to the SSH data based on whether or not a varying threshold was exceeded, and subsequently saving the

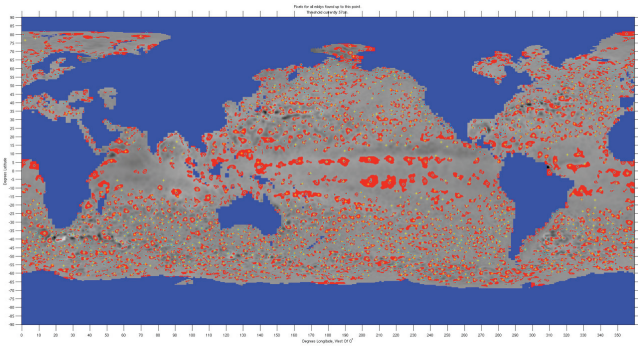

Figure 2: Global cyclonic eddy-like features identified in a single SSH time frame. We identify several thousand eddy-like features in any time frame, many of which will be discarded after significance tests. The goal is to track eddies by connecting these nearly ubiquitous features with their corresponding features in subsequent time frames.

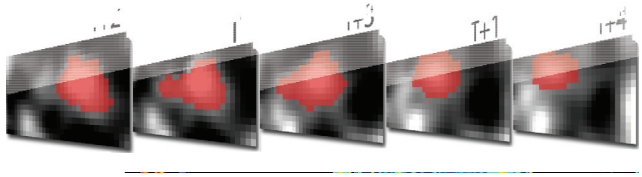

Figure 3: A moving eddy as identified in five successive time frames of SSH satellite data. Note how the feature changes size and shape from on frame to the next due to noise in the SSH field.

eddy-like connected component features that remain after thresholding. The identified features are further pruned based on physically-consistent criteria that define eddies (Chelton, Schlax, and Samelson 2011; Faghmous et al. 2012). Figure 2 shows the ubiquitous cyclonic eddy features identified in a single SSH snapshot. Each snapshot contains several thousand eddy features. However that number is often reduced by a variety of significance tests to remove spurious discoveries.

Once eddy-like features have been identified in all time frames, they need to be tracked across time. Figure 3 shows an example of an eddy identified in five successive SSH frames. While tracking a single object is trivial, tracking multiple user-defined features presents unique challenges both conceptually and computationally. First, since the objects being tracked are not self-defined with clear boundaries like physical objects (*e.g.* car, ball, *etc.*) the very notion of an object is subject to interpretation and errors. This can be seen in Figure 3, where the size and shape of the feature changes across time due to noisy measurements. Second, given the multitude of objects in any given frame, the computational resources required to maintain a large number of potential tracks grows exponentially. Finally, splitting and merging are regular behaviors of ocean eddies that are not common concepts in the traditional object tracking literature.

The majority of eddy tracking algorithms employ a computationally modest, yet limited approach where a feature is connected to the nearest feature in the subsequent time frame. While this approach gives reasonable results

in eddy tracking applications (Chelton, Schlax, and Samelson 2011), its greedy nature means that there will often be cases where it performs sub-optimally, especially in noisy or cluttered environments (Bar-Shalom and Li 1995; Blackman 2004).

Based on the following overview, we propose the following problem definition and present the methods used to solve such a problem in the following section.

Problem Definition

Given T frames containing eddy features identified in an independent detection step, with each frame i having N_i eddy features. Extract globally-coherent eddy tracks, where an eddy track is a group of eddy features, satisfying the following constraints: (1) each track has at most one feature from consecutive time frames; and (2) each feature is associated with at most one track.

3 Methods

We implemented *local nearest neighbor* assignment (LNN): the most widely used method in ocean eddy tracking (Chelton, Schlax, and Samelson 2011). We also implement *multiple hypothesis assignment* (MHA) adopted from Blackman (2004), where each feature is associated with multiple potential tracks.

3.1 Local Nearest Neighbor (LNN)

In the LNN scheme, for each eddy feature identified at time t, the features at time $t+1$ are searched to find the closest feature within a pre-defined search space based on the theoretical distance an eddy can travel during the period between two successive time frames. In the event that a feature at time $t+1$ is closest to two features from the previous step, it is assigned to the first encountered feature. This causes the resulting tracks to depend on the scanning order of the feature set.

3.2 Multiple Hypothesis Assignment (MHA)

The main difference between MHA and LNN is that unlike LNN, tracks are not formed at every time step. Instead, features are assigned to multiple plausible tracks and uncertain decisions are deferred until more data are available to make a relatively unambiguous track assignment. We use a *track tree* to maintain all possible tracks that be can be constructed starting from any eddy (shown in Figure 4). Each path along the track tree is a potential track starting from its root.

In an MHA setting, every track has three outcomes: initiation, expansion, or termination. At any time t, all potential expansions and terminations for the active tracks at $t-1$ are entered in the track tree. Additionally, all features identified at time t initiate potential new tracks by creating new track trees with the new features as the root.

Figure 4 demonstrates how MHA maintains all plausible hypotheses across 3 time-steps and 7 eddy features. In this scenario, there are two eddies moving in space and time (e_1,e_3,e_6) and (e_2,e_4,e_7), and a spurious eddy feature e_5. At t_1 the only possibility is to initiate two track trees with e_1 and e_2 as roots. At t_2, three eddy features emerged and they

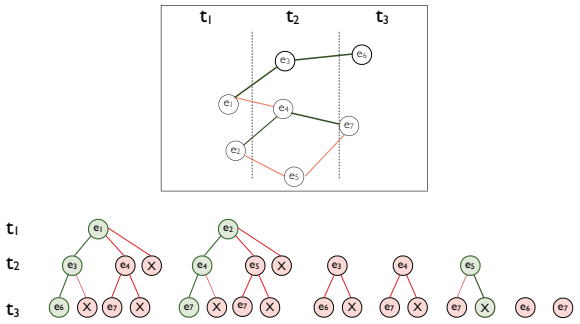

Figure 4: An illustrative example of how MHA maintains all plausible tracks in memory. *Top panel:* A two-dimensional spatio-temporal view. Each panel represents one time step. The real tracks are (e_1, e_3, e_6), (e_2, e_4, e_7), and e_5 and are emphasized by green edges. The light red edges are plausible tracks that are considered by MHA but ultimately discarded. *Bottom panel:* the corresponding track trees for all possible tracks based on the data in the top panel. The tracks selected are highlighted in green. Abandoned tracks are in red. "End" nodes are denoted by "X". Note: for simplicity the scoring function, lookahead, and self-learning features are not demonstrated in this example. Figure best seen in color.

extend the trees created by e_1 and e_2 as well as initiate new trees. Moreover, the tracks created by e_1 and e_2 may end at t_2, as denoted by the "X" nodes in Figure 4. Finally two more eddies appear at t_3 and they too are associated with existing trees as well as initiate new ones (denoted by the rightmost single-node trees e_6 and e_7).

While this approach may be more accurate than LNN, the enumeration and maintenance of all possible tracks grows exponentially. To reduce the problem's complexity, we use *gating* and *pruning* functions (Reid 1979; Kurien 1990). Gating ensures that only eddy features that are within a maximum search distance from existing tracks are considered as potential extensions. The gating distance, g, varies based on the location of the track, since the size and distance eddies travel vary uniformly as a function of latitude (Fu et al. 2010). An example of gating can be seen in Figure 4, where e_5 is not considered as e_1's extension because e_5 is outside e_1's gate.

Furthermore, we employ *N-scan pruning*, where we solve for any ambiguities across all track trees by finalizing the assignment of eddies that appeared during the past N-steps. Hence, every N steps, we reduce the number of paths along each track tree to at most one. To do so, we assign every edge in each track tree a reward r = g - $d(e_i^t, e_j^{t+1})$ if we extend a track at time t such that the distance between the connected features at time t and $t+1$ is $d(e_i^t, e_j^{t+1})$. That is, we give higher scores to tracks that extend through the nearest feature in the search space.

To make final track assignments, we sum the scores of each edge along a path from time $t-N$ to t and select the highest scoring path from each track tree. If two highest scoring paths from different trees share a node (*i.e.* they are ambiguous), then we select the track with the highest total score.

Referring back in Figure 4, If we adopt a 3-scan pruning procedure ($N=3$), all track trees must be pruned every 4 steps. In this case, each path along every tree will be scored by

summing the rewards r of all edges along the paths between times $t = 1$ and $t = 4$. The only tracks remaining from the pruning phase are the highest scoring non-conflicting tracks across all trees. One example of conflicting tracks are (e_1,e_3,e_6), which is the highest scoring track in the left-most tree, and (e_3, e_6), the highest scoring track in the third tree from the left. They are conflicting because they share e_3 and every feature can only be part of at most one track. In case of conflict, MHA will select the track that has the highest total reward, which in this case is the longest track (e_1,e_3,e_6). Therefore, the tracks (e_1,e_3,e_6) and (e_2,e_4,e_7) are non-conflicting tracks and together have the globally optimal reward. If new features are identified at $t = 4$ (not shown), then the only possible extensions are for the first and second trees along the highest scored path (green path) or to extend e_6 and e_7. All remaining red paths and trees are pruned and are no longer eligible extension.

Using the same example, LNN is unable to recover the proper tracks in Figure 4. At time t_2, LNN will assign e_1 and e_2 to the closest features which are e_4 and since e_4 is already paired with e_1, e_2 is associated with the second closest feature, $e5$. At t_3, LNN will continue to diverge for the proper path, by assigning e_4 to e_7 and terminating e_5. This will result in an incorrect final output of (e_1,e_4,e_7), (e_2,e_5) and (e_3,e_6).

To address the noise in SSH data, MHA extends the traditional multiple-hypothesis tracking algorithm to account for noisy observations, where features contain varying degrees of uncertainty. The two main extensions of MHA that allow it to handle noisy data are: *lookahead*, where we do not terminate tracks that were not extended for a single time step in the event that an eddy "disappeared". The second feature is unsupervised self-learning, where MHA autonomously detects an error in the feature identification phase and takes a corrective measure *a posteriori*.

3.3 MHA Lookahead

A common challenge in tracking eddies is that some eddies might be left unassociated from one time-step to the next due to eddies temporarily "disappearing" as a result of noise and sampling errors. CH11 attempted to address this problem by allowing features to not be paired with a subsequent feature for 2-3 time steps in the hope that it will be associated to a track at a later time. However, due to LNN's localized greedy nature the "procedure for tracking temporarily lost eddies were disappointing. The resulting eddy trajectories often jumped from one eddy to another" (Chelton, Schlax, and Samelson 2011). Thus, while this lookahead feature is highly desirable, it was abandoned in the most comprehensive study to date, as it was preferable to shorten eddy tracks than to incorrectly associate eddies to incorrect tracks. MHA implements the lookahead feature by assigning a "missing" node to tracks that were not associated with an eddy instead of terminating them. If two consecutive "missing" nodes are within the same path, then it is terminated. MHA avoids jumping tracks because it maintains all plausible future tracks and will likely choose the more consistent track since it has more data available than LNN does.

3.4 MHA Unsupervised Self-Learning

One fundamental difference between our method and existing ones is its ability to identify errors from the eddy detection phase and correct them autonomously. The most common type of misidentification is that of artificially merged eddies. Due to the noisy SSH data, features that are in close proximity are susceptible to being labeled as one large eddy (Chelton, Schlax, and Samelson 2011). Faghmous et al. (2012) addressed such artificial merges by introducing a convexity measure on the features detected to ensure that only the most compact features are selected, as merged features tend to have abnormal shapes. Nonetheless, due to the global nature of the autonomous identification scheme, some clusters of eddies may still be merged as a single large feature. Assume that at time t there are three eddies e_1, e_2, and e_3 with areas a_1, a_2, and a_3 respectively. When a new eddy new_e appears at time $t+1$ it is automatically associated with e_1, e_2, and e_3. If new_e is related to either existing eddy it should be roughly their size. If new_e is a merged eddy, then its area would be significantly larger than *all* existing eddies. Therefore, if MHA detects a new eddy where all of its associated predecessors in the track tree are at least 1/2 of its size, we flag the new eddy as a potential merge and apply the eddy identification algorithm described in (Chelton, Schlax, and Samelson 2011; Faghmous et al. 2012) to break up the large eddy into smaller ones. If the self-learning procedure returns new eddy features, we then remove all nodes that contained the artificially large eddy from all track trees and extend all active tracks with the new features. This will cause the trees' breadth to increase by $K - 1$ branches where K is the number of newly identified features. The new features will also initiate K new track trees.

The unsupervised self-learning feature is only possible thanks to the multiple hypothesis nature of MHA. There are significant conceptual challenges in extending unsupervised self-learning to an LNN framework. Assume that an eddy feature e_1 is associated to a significantly larger feature e_2 in the following time frame. In order for LNN to label e_2 as an artificial merge with high probability, it must consider all possible feature pairings in the search space to ensure that there is no other feature, e_i that is of the same size as e_2 and could be attached to it. Even then, LNN could not account for the scenario where e_2 is simply a new large eddy that just initiated and will continue during the subsequent time frames.

3.5 Computational complexity

Given M, the maximum number of objects in any time frame, and T time-steps, LNN would compare the distance between every other object leading to M^2 comparisons. Therefore, the worst-case runtime complexity of LNN is $O(M^2T)$.

For MHA, let K be the number of hypotheses maintained for consideration at any given time-step. In the worst case, $K = M^T$ where MHA would store all possible tracks, however using N-scan pruning, MHA only stores the N-most hypotheses resulting in $K = M^N$. The lookahead and self-learning features add additional costs to the algorithm. However their running time cannot be explicitly bound but in most

reasonable cases it should be no more than $O(M)$. As a result, the worst-case runtime complexity of MHA is $O(M^{N+1}T)$.

4 Results

We identified eddies globally in weekly unfiltered SSH data from October 1992 to January 2011 using the procedure described in (Chelton, Schlax, and Samelson 2011; Faghmous et al. 2012). We used the Version 3 dataset of the Archiving, Validation, and Interpretation of Satellite Oceanographic (AVISO) which contains 7-day averages of SSH on a $0.25°$ grid. Given the absence of eddy baseline data or "ground truth", evaluating and comparing methods is a notable challenge. The majority of eddy tracking studies focus on aggregate results such as track lifetimes and distance propagation. An evaluation that is more relevant to our audience would compare the performance of each algorithm at detecting tracks. Ideally, one would use "ground truth" data where the eddy tracks are known in advance and test how well each method recovers such tracks under varying conditions. One way to generate such data would be through a numerical simulation (*i.e.* ocean model) with idealized eddies as ground truth and then gradually add noise. However, such simulations are computationally expensive and require sophisticated physics-based models to simulate eddies and their trajectories. An alternative would be to use filed studies data, where floats are dropped in the ocean and subsequently tracked and eddies are identified when the float rotate while translating (*i.e.* the float is moving along the translating eddy). However, such data make up a small sample size and are not sufficient to significantly differentiate between the two methods. To address these limitations, we designed an experiment to test the relative robustness of each method to noisy measurements; a significant concern in SSH data.

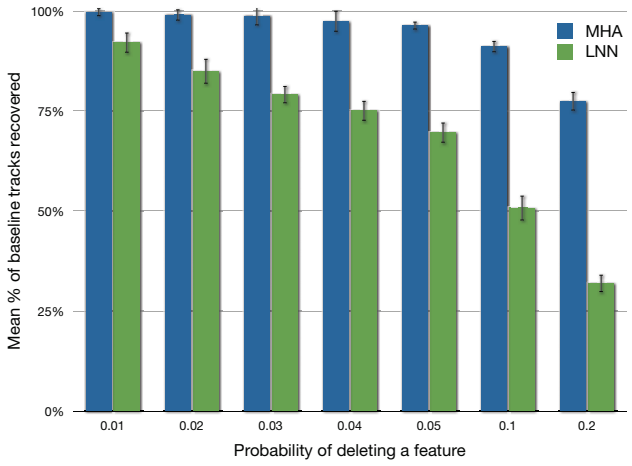

Figure 5: The mean percentage of baseline tracks recovered by LNN and MHA as a function of the probability deletion (p). At every timestep, each feature may be removed from the data with a probability p. Each probability level was simulated 50 times and the results show the mean of the 50 simulations. The error bars denote standard deviations.

4.1 Sensitivity To Noise

There are three ways that noise can affect automatic eddy monitoring procedures. First, a spurious connected component feature may appear for a single time-step before disappearing. Second, noise may cause uncertain feature boundaries and, as a result, shifted feature centroids. Finally, features might disappear temporarily either due to limitations in the identification scheme, or because of perturbations in the SSH field make the feature unidentifiable. The first type of noise is handled by discarding features that do not persist over four weeks. The second can be handled by smoothing or taking a weighted centroid. The final type of noise is the most vexing. As highlighted earlier, disappearing features is a considerable challenge that CH11 could not address. As a result, we focus our evaluation on the resilience of LNN and MHA to disappearing features.

To test each algorithm's ability to handle missing features, we built a baseline dataset and randomly removed features from that data. We then computed how many of the original baseline tracks were recovered despite the noise. We constructed the baseline dataset by running both LNN and MHA on the features identified in the South Pacific ocean for the years 2005 and 2006. Only tracks that lasted at least 4 weeks and perfectly matched between both methods were selected for inclusion in the baseline dataset. We simulated the event of a feature temporarily disappearing by randomly removing features in the baseline dataset based on a variable "deletion" probability p. To account for a variety of conditions that lead to disappearing eddies, we vary p from 0.01 to 0.2 and run both LNN and MHA algorithms on the noisy dataset. Figure 5 shows the results of the experiment. As it can be seen, LNN's recovery rate degrades much faster than that of MHA. It should be noted that a track may have multiple features deleted and that longer-lived tracks are more likely to be missed since there is an increased likelihood of multiple features being removed from longer tracks. However these issues are the same for either procedure, therefore not significantly affecting the results. This experiment empirically shows that MHA is significantly more robust to noise relative to LNN, even in extreme conditions (*e.g.* $10 - 20\%$ probability of missing a feature).

4.2 Impact of Missing Eddies on Track Statistics

LNN's sensitivity to missing eddies can have a significant impact on reported track durations, as eddies disappearing for a single time step would cause tracks to terminate prematurely when the eddy is still moving. Figure 6 shows an example where MHA performs better than LNN in the case of an eddy "disappearing" for one time-step. Initially, both methods track the eddy identically. Yet, at time t_3 the eddy identification method is unable to find an eddy and since LNN does not implement lookahead, the track was terminated. MHA, however, was able to recover the full track by allowing the track to be unassociated for a single time-step and then continue tracking once the eddy reappeared.

To further quantify the effect of such premature terminations on track statistics, we repeated the noise experiment reported earlier on all features in the South Pacific between

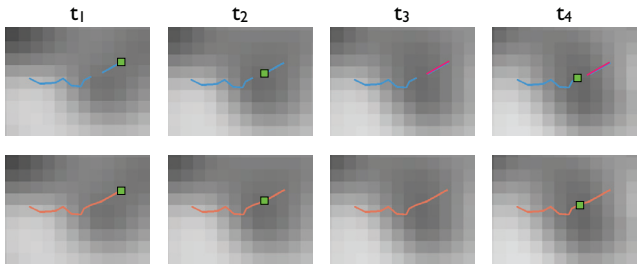

Figure 6: An example of how MHA is able to recover tracks even when eddies temporarily "disappear." The eddy is moving westwardly (from right to left). Top (from left): An eddy centroid (green square) and its path as detected by LNN with no lookahead ability. At t_3 the eddy detection algorithm loses the eddy for one time step as it reappears in t_4. However, without the lookahead feature, LNN breaks up the long track into two small tracks as indicated by the different colored tracks. Bottom (from left): the same eddy tracked with MHA. Although the eddy is lost at t_3, MHA successfully recovers the track once the eddy reappears at t_4.

2005 and 2006 (not just the baseline data). Due to space limitations, we only analyze one deletion probability of $p = 0.04$ (the median of the probabilities from the previous experiment). Figure 7 shows the mean cumulative track lifetimes for 50 simulations where features have a 4% probability of temporarily disappearing. Although both methods identify nearly the same number of total tracks, LNN identifies 20% less tracks that live 11 weeks or longer. That is a significant difference given that long-lived eddies play a fundamental role in transporting water across ocean basins.

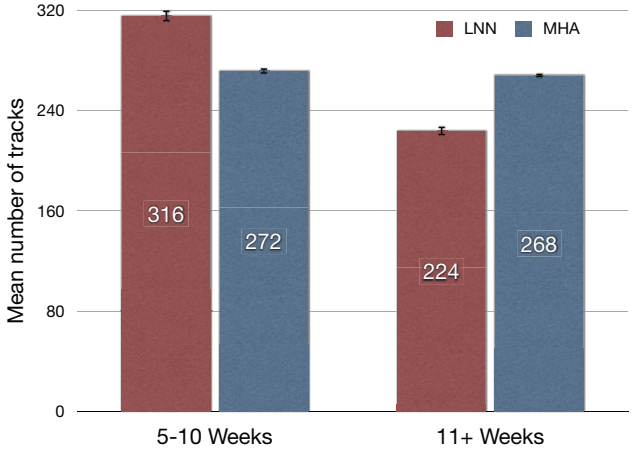

Figure 7: Mean cumulative track lifetimes for LNN and MHA tracks from the noise experiment with a probability $p = 0.04$ of a feature temporarily disappearing. MHA identified 20% more long-lived tracks (11+ weeks) than LNN. The simulation was run 50 times and the error bars denote standard deviations.

4.3 Impact of Self-Learning on Artificial Merges

Figure 8 shows two examples of unsupervised self-learning. In both cases, artificially large eddies were saved in the identification step, but MHA was able to flag such errors autonomously and corrected them by splitting the large features into more accurate smaller eddies. The large panels (A) in Figure 8 highlight the artificially merged eddies over a background of greyscale SSH data. These merges were identified by MHA in an unsupervised manner where it would flag any features that had *all* potential tracks associated with it be significantly smaller in previous steps.

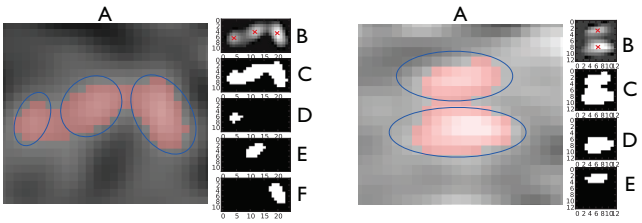

Figure 8: Two examples of artificially large eddies being merged as a single eddy and MHA's ability to identify such errors. Panel A denotes the actual SSH data with the large eddy highlighted in red. The smaller vertical panels on the right (B-F) show the binary view of the merged eddy and the corrected data. Panel B shows the actual greyscale SSH data inside the eddy, with the multiple minima marked by red crosses – an indication of an artificial merge. The C panel shows the pixels associated with the eddy as saved by the identification scheme. The bottom two panels (D-F) show the new smaller eddies after data correction.

The blue ellipses in panel A of Figure 8 highlights the merged features in red. Panels B and C show the features as saved in the eddy identification phase in a grayscale and binary forms respectively. The crosses in panel B indicate the local minima, which are centroids of the actual eddies that were erroneously merged. Panels D-E show the resulting eddies once the data correction procedure was performed. In both cases, the separation occurred along the proper direction as denoted by the major axis of the blue ellipses in panel A. These examples demonstrate how the unsupervised self-learning feature effectively leverages the spatio-temporal context of the data to make accurate corrections to the unsupervised eddy detection phase. Such an extension can be used in other multiple hypothesis tracking settings where the objects have varying degrees of uncertainty.

One common sign of an artificial merge is multiple extrema within a single feature. That is because, physically, eddies should only have a single extrema. Using MHA's self-learning feature we were able to autonomously identify 2185 artificial merges within the South Pacific ocean for years 2005 and 2006. Of those potential merges, 1682 features had multiple extrema. An extrema was defined as a pixel whose value was greater/less than all of its 5×5 neighbors. We were able to successfully break 1058 of the reported merges, resulting in a significant increase in feature counts.

5 Summary & Future Work

We presented MHA: a multiple hypothesis tracking-inspired eddy monitoring application. MHA extends the traditional multiple-hypothesis tracking approach to handle noisy observations and introduced an unsupervised self-learning feature that refines objects *a posteriori*. Despite the lack of "ground

truth" data to evaluate our algorithm, we presented several experiments and case-studies that demonstrated that our method addresses two major shortcomings of the most widely used method in the eddy tracking literature. Furthermore, MHA is an example of how incorporating the data's spatio-temporal context can improve the accuracy of feature identification and tracking algorithms.

Our MHA algorithm is able to use its multiple hypothesis nature, along with the data's spatio-temporal context, to take corrective measures on a previously independent eddy identification phase. As far as we are aware, this is the first approach in the eddy tracking literature to do so. Furthermore, our method is more robust to noise as it successfully implements a lookahead feature that makes it less likely to break-up tracks if observations temporarily disappear.

Additional analysis of MHA's performance based on various factors (*e.g.* eddy density, propagation speed, *etc.*), as well as self-learning's impact on track statistics will be the subject of future work. MHA could also be extended to handle the evolutionary nature of eddies forming and dissipating by adopting MCMC techniques from biological sciences (Smal et al. 2008). While MHA's pruning heuristic is based on theoretical eddy dynamics, less rigid heuristics such as the joint probabilistic data association (JPDA) method (Bar-Shalom and Tse 1975; Bar-Shalom and Li 1995) may be more accurate. Finally, now that it has been shown that MHA is a more robust alternative to LNN, concepts such as tracks splitting and merging can be incorporated in the MHA scheme by relaxing the constrain of having an eddy be part of a single track at most. Such an addition may provide more accurate information about ocean dynamics.

6 Acknowledgements

The authors would like to thank three anonymous reviewers whose suggestions significantly improved the manuscript. This research was funded by a University of Minnesota Doctoral Dissertation Fellowship, the Planetary Skin Institute, NSF Grant IIS-0905581,and an NSF Expedition in Computing Grant (IIS-1029711). Access to computing resources was provided by the University of Minnesota Supercomputing Institute.

References

Bar-Shalom, Y., and Li, X. 1995. *Multitarget-multisensor tracking: principles and techniques*. YBS Publishing, Storrs, CT: University of Connecticut.

Bar-Shalom, Y., and Tse, E. 1975. Tracking in a cluttered environment with probabilistic data association. *Automatica* 11(5):451–460.

Blackman, S. 2004. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE* 19(1):5–18.

Caldeira, K.; Wickett, M.; et al. 2003. Anthropogenic carbon and ocean ph. *Nature* 425(6956):365–365.

Cham, T., and Rehg, J. 1999. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE.

Chelton, D.; Gaube, P.; Schlax, M.; Early, J.; and Samelson, R. 2011. The influence of nonlinear mesoscale eddies on near-surface oceanic chlorophyll. *Science* 334(6054):328–332.

Chelton, D.; Schlax, M.; and Samelson, R. 2011. Global observations of nonlinear mesoscale eddies. *Progress in Oceanography*.

Cox, I., and Hingorani, S. 1996. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18(2):138–150.

Dong, C.; Nencioli, F.; Liu, Y.; and McWilliams, J. 2011. An automated approach to detect oceanic eddies from satellite remotely sensed sea surface temperature data. *Geoscience and Remote Sensing Letters, IEEE* (99):1–5.

Faghmous, J. H.; Styles, L.; Mithal, V.; Boriah, S.; Liess, S.; Vikebo, F.; Mesquita, M. d. S.; and Kumar, V. 2012. Eddyscan: A physically consistent ocean eddy monitoring application. In *Intelligent Data Understanding (CIDU), 2012 Conference on*, 96 –103.

Fernandes, A. 2008. Identification of oceanic eddies in satellite images. *Advances in Visual Computing* 65–74.

Fu, L.; Chelton, D.; Le Traon, P.; and Morrow, R. 2010. Eddy dynamics from satellite altimetry. *Oceanography* 23(4):14–25.

Han, M.; Xu, W.; Tao, H.; and Gong, Y. 2004. An algorithm for multiple object trajectory tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, I–864. IEEE.

Hoegh-Guldberg, O., and Bruno, J. 2010. The impact of climate change on the worlds marine ecosystems. *Science* 328(5985):1523–1528.

Kurien, T. 1990. Issues in the design of practical multitarget tracking algorithms. *Multitarget-Multisensor Tracking: Advanced Applications* 1:43–83.

Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B.; et al. 2003. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, volume 18, 1151–1156. LAWRENCE ERLBAUM ASSOCIATES LTD.

Nieto, J.; Guivant, J.; Nebot, E.; and Thrun, S. 2003. Real time data association for fastslam. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, 412–418. IEEE.

Oh, S.; Russell, S.; and Sastry, S. 2009. Markov chain monte carlo data association for multi-target tracking. *Automatic Control, IEEE Transactions on* 54(3):481–497.

Pegau, W.; Boss, E.; and Martínez, A. 2002. Ocean color observations of eddies during the summer in the gulf of california. *Geophysical Research Letters* 29(9):1295.

Poore, A. 1995. Multidimensional assignment and multitarget tracking. *Partitioning Data Sets. DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 19:169–196.

Reid, D. 1979. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on* 24(6):843–854.

Smal, I.; Draegestein, K.; Galjart, N.; Niessen, W.; and Meijering, E. 2008. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. *Medical Imaging, IEEE Transactions on* 27(6):789–804.