Research on mesoscale eddy-tracking algorithm of Kalman filtering under density clustering on time scale

Ji-Tao Li and Yong-Quan Liang

ABSTRACT
This article proposes the tracking algorithm based on density clustering of time scale and mesoscale eddy of Kalman filtering using the fused SLA data of altimeter. Firstly, the definitive density clustering based on time scale discovers the potential association pattern between data, and screens out the data set of mesoscale eddy trajectory. With regard to the data set with time scale conflict, it analyzes the Kalman filtering, eliminates the noise points and obtains the correct mesoscale eddy trajectory. Secondly, it turns the tracking process into an algorithm that supports batch processing by applying the data processing method to the mesoscale eddy-tracking algorithm, which solves the problem of single serialization and high time and space complexity of the traditional tracking algorithm. Based on the algorithm, this article selects the experimental data of the South China Sea for the mesoscale eddy-tracking test. The experiment turns out that the algorithm can better reveal the life course of mesoscale eddy and evolution rule of physical oceanography according to spatial scale, amplitude and eddy duration, etc.

Introduction
Mesoscale eddy, as the transmitter of energy and material exchange in the ocean, with ocean characteristics of the closed interval. The kinetic energy brought by it comes from the influence of Coriolis Force of the earth, which forms a counterclockwise cyclonic eddy and a clockwise anticyclonic eddy (Okubo, 1970). The research on the mesoscale eddy in the ocean mainly includes the identification, observation and tracking of mesoscale eddy, which is of great significance to military oceanography, ship transportation and fishery production.

This article obtains the global sea surface height based on the satellite altimeter data, and obtains SLA (Sea level anomaly) data after data correction and data merging. The mesoscale eddy on this map at this time is manifested as the cyclonic eddy filled with cool color and the anticyclonic eddy filled with warm color. With regard to the research on the tracking of mesoscale eddy, most of the existing research ideas are carried out based on the tracking method of closest point. This method solves the error problem caused by the disappearance of the mesoscale eddy by setting the threshold, and selects the node closest to the precedent node as the subsequent node for the circular traversal of all documents (Nencioli, 2010). Based on the above method, a large number of experiments can be applied to the tracking analysis of the mesoscale eddy in the seas around the world. The development and improvement in subsequent methods are also based on this.

Doglioli uses wavelet transform method to trace the Agulhas Current between the Atlantic Ocean and the Indian Ocean, and analyzes SSH (Sea Level Height) data of altimeter that changes continuously through wavelet transform (Doglioli et al., 2007). Meanwhile, this article also applies wavelet transform to the same eddy-tracking analysis of mesoscale data, SST (Sea Surface Temperature) data and salinity data in the same area. The tracking method is the classical method of nearest neighbors, which selects the mesoscale eddy characterized by the shortest time and minimum vertical height difference as the subsequent eddy by comparing the nearest neighbors of time scale and vertical height scale. The algorithm selects the time window of 2 days, the vertical height difference between adjacent mesoscale eddies is less than 100 m, and the threshold-less than 1000 m. The method has added restricted conditions, but the time window is too short and the intelligence degree is too low, resulting in low accuracy. Chaigneau has proposed a dimensionless distance way as the way to determine eddy distance to track the eddy (Chaigneau et al., 2008), and added distance scale, eddy radius, vorticity and EKE to such dimensionless distance. In this way, it can compare the eddy properties from more attributes. Based on such dimensionless measuring method, Chaigneau has carried out an
eddy-tracking experiment on the Peru seas in the South Pacific (Chaigneau et al., 2009). The algorithm has achieved a series of effects, but the practical research shows that its shortcoming is that the definition of threshold coefficient needs sensitivity testing. Chelton has selected the method of nearest neighbors for eddy-tracking test in his article (Chelton et al., 2011), but added the determination of the range of motion and prevailing cycle of eddy in the tracking process, of which the determination method is that the range of motion within the life cycle of eddy is less than $10^\circ$, and analyzed the eddy with a continuous cycle of more than 16 weeks. Yi used the method of nearest neighbors in this article and conducted the tracking analysis of mesoscale eddies combined with Kalman filtering method (Yi et al., 2017). Yi introduced the idea of machine learning into the traditional mesoscale eddy-tracking method for the first time, and analyzed the probability before and after the time quantum in the Bayes way, which effectively reduced the error rate in the tracking process. Through variance control, Yi reduced the error rate within 0.2% under the premise that eddy disappeared after 3 days.

To sum up, there are still some problems in the existing algorithm. The traditional mesoscale eddy-tracking algorithm is a typical serial method, which cannot support batch processing. The initial judging criteria of eddy tracking is very complex, which requires a larger storage space for storage in the intermediate process, during which the algorithm has a high complexity and a long running time. Even in the process of tracking and processing using filtering method, there are a series of problems brought by the serial method that need to be solved. Based on the consideration of above problems, this article proposes a kind of mesoscale eddy-tracking algorithm based on the density clustering on time and space scales and Kalman filtering. In considering that density clustering is suitable for all kinds of multi-shape clusters, meanwhile the trajectories of mesoscale eddy are anomaly, and the intermediate result of clustering could be filtrated, which explains the algorithm is able to be optimized by reset of parameters. Based on these advantages, density clustering could be used to establish the access between data through screening and filtering steps on time and space scales concerning solving the single serialization problem in the tracking process of mesoscale eddy, discovers the potential links between data through clustering analysis and establishes the position point set of the movement trajectory of mesoscale eddy. Due to the error of calculation and the observation error, the potential trajectories may have reentrance phenomenon, which is unacceptable. Later, this article uses the Kalman filtering method to eliminate concentrated noise points and obtains a more accurate movement trajectory of mesoscale eddy, because Kalman filtering method is one of the most convenient methods, it has low time cost, easy to realize and high efficiency of operation. The algorithm in this article compares the traditional algorithms that can be used for serial operation only, as a result of which the operation efficiency has been improved obviously, and batch processing ability of the algorithm has been greatly enhanced. The process of data processing is based on the whole data set. Different from the traditional algorithms that seek certain fixed two-day data only, the method is more stricter with the data set, and reduces the misjudgment rate on this basis by introducing the method of multidimensional filtering. The experiment turns out that the algorithm in this article can deal with the “disappearance” and “jumping” problems of mesoscale eddy more effectively, and avoid the mistake that a path is cut into two or more paths due to the two problems above, which greatly improves the accuracy rate of tracking.

**Theoretical model**

**Experimental data (altimeter data)**

The satellite altimeter data used in this article is the fusion data of SLA of multisource altimeter distributed by AVISO. The SLA data source comes from altimeter data of satellites such as TOPEX/Poseidon, Jason-1, Jason-2, ERS-1, ERS-2 and Envisat. The spatial resolution is $0.25^\circ \times 0.25^\circ$, which fuses the altimeter data of at least two satellites, so that mutual calibration can be achieved between data of multiple satellites, which ensures the accuracy and consistency of the fused products. AVISO provides altimeter data from 1 January 1993 to 30 September 2016, and also provides the position, amplitude, kinematic velocity and movement trajectory of the eddy whose life cycle is greater than 2 weeks.

**Traditional tracking algorithm and existing problems**

The tracking algorithm of traditional mesoscale eddy is shown below (Sun et al., 2017):

1. Data initialization: Select the start time $t$ as the start time of the tracking algorithm.
2. Search process: Record all mesoscale eddies at the moment $t$, and search for similar mesoscale eddy within the moment $(t + 1)$ as the succession of corresponding mesoscale eddy at the moment $t$. The judgment method is to set the radius threshold as 50 km.
3. Selection process: After setting the threshold, enter the screening process. Within the moment $(t + 1)$, select the one with the minimum
difference within the range of threshold as corresponding mesoscale eddy at the moment \( (t + 1) \). Since the mesoscale eddy may disappear in the life cycle, continue to search for the mesoscale eddy of next moment if corresponding mesoscale eddy is not found at the moment \( (t + 1) \). If the mesoscale eddy is still not found at the moment \( (t + 5) \), the mesoscale eddy is deemed to have disappeared, and the life cycle comes to its end.

(4) Renewal process: If corresponding precedent node is not found within the moment for certain mesoscale eddy at the moment \( (t + 1) \), the node shall be considered as the start eddy for tracking mesoscale eddies, and then repeat the selection process.

According to the steps mentioned above, main problems solved realized by the traditional tracking algorithm are shown below:

(1) Solve the problem of similarity measurement of mesoscale eddies at moment \( (t + 1) \) and moment \( t \) through judgment of wave height range and range restriction of the distance of eddy core.
(2) If the algorithm is used to distinguish “jumping” and “disappearance” problems in the intermediate process of eddy, the solution is to search for the eddies within the range of 50 km. If similar eddies cannot be found in 5 days, the eddies are deemed to have disappeared.
(3) If the algorithm is used to deal with the storage of eddies in the life cycle, the solution is to find the initial point of eddy core of the precedent node.

The advantage of the traditional algorithm is that its algorithm thought is simple and is in line with the normal thinking steps of the existing non-artificial intelligence method. However, the intelligence of the algorithm needs to be improved. In addition, the implementation of the algorithm requires more time and high space complexity. Non-global data set processing could lead a problem that results may get into the local optimal, or different parameter initialization may lead a uncertainty of accuracy. Lastly, the algorithm is weak in batch processing. At the same time, the consideration on time and space scales is restricted to the basic judgment conditions only, so it cannot be effectively integrated into the tracking process.

**Steps proposed in this article**

To improve the accuracy and computational efficiency of the traditional mesoscale eddy detection model, this article introduces the density clustering method in machine learning after correction and adaptation into the tracking process of movement trajectory. Restricted by time scale, this article classifies the data points with potential association pattern as the same category after the clustering process, and then obtains the subset of data points that belong to different labels. The data in the subset are ranked based on time in ascending order, and then the fuzzy movement trajectory of its probability is obtained. In the next process of model building, the following judgment rules are proposed: If there is no time scale conflict in the subset, the path is considered as the final trajectory path. If not, proceed with Kalman filtering processing, filter out noise points and obtain the correct trajectory path. It is important to note that the algorithm in this article does not need to obtain the results through traversal, which greatly reduces the computational complexity. The results shall be integrated after the said two steps are completed, and the tracking process of algorithm trajectory comes to its end.

**Density clustering based on time scale**

To solve various problems mentioned in the traditional algorithm, this article adopts density clustering based on time scale. The good quasi-adaptivity and noise insensitivity of density clustering are widely used in the work. Among many algorithms of density clustering, the most commonly used algorithm is the DBSCAN algorithm (Ester, 1996). There are a lot of improvements derived on this basis on specific areas, such as ST-DBSCAN (Birant & Kut, 2007) for time and spatial data analysis and fast processing MR-DBSCAN for MapReduce (He et al., 2012). At present, the density clustering method is widely used in data mining (KDD). Its unsupervised feature does not require the training in advance of the training set, which discovers the potential links between data by virtue of the preliminary definition and is widely used in data compression (Bradley et al., 2002), abnormal point detection, image segmentation (Manavalan & Thangavel, 2011) and other fields.

The coefficient definition of DBSCAN requires sensitivity test and needs to understand the data in advance, but this shortcoming can be avoided well when the data set attributes are fixed. The density clustering used in this article uses the single way of distance scale only as compared to traditional density clustering, and uses time scale to intervene the clustering process, aiming to effectively avoid inaccurate clustering results caused by the overlapped tracking paths.

Definitions in density clustering are shown as follows:

(1) The distance between points in data set (DB) is \( \text{Dist}(p, q) \), where \( p \) and \( q \) belong to DB
\[
\text{Dist}(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}
\]  

(1)

Based on the time scale, it is necessary to determine whether the eddy time of two points is greater than 5 days. If greater than 5 days, then the distance between two points is positively infinite. If the difference between the two eddies is too large, the distance between two points will also become positively infinite.

\[
D(p, q) = \begin{cases} 
\text{Dist}(p, q), & \text{abs}(p - q) \leq 5 \\
\inf, & \text{abs}(p - q) > 5 
\end{cases}
\]  

(2)

(2) For Point \( p, p \in DB \). If the circle of the given object \( p \) with the radius of epsilon, it at least contains MinPts samples, then the points within the radius are deemed to belong to the data set \( \epsilon - \text{nbhd}(p) \).

(3) For Point \( q \) in data set \( DB \), if the distance between Point \( q \) and Point \( p \) is smaller than epsilon and belongs to the data set \( \epsilon - \text{nbhd}(p) \), Point \( q \) is deemed to be directly accessible for Point \( p \).

(4) For \( p \) and \( q \), if it is accessible from \( p \) to \( q \) indirectly in the samples in a directly accessible way through sample \( x_1, x_2 \ldots x_n \), the density is deemed as accessible. It can be obtained from the asymmetry of density accessibility that the density accessibility meets symmetry.

(5) For Point \( p \) and Point \( q \), it is deemed as density connection that meets symmetry if the two points are mutually accessible in terms of density.

With regard to the density-based clustering for a data set \( C \): 1) For any \( p \) and \( a \) with the same class label, then \( p \) and \( a \) are connected in terms of density; any \( p \) belongs to Class \( C \), \( q \in DB \), \( q \) is included Class \( C \) if \( p \) and \( q \) are connected in terms of density. If some point cannot be classified into any class, it is known as a noise point. In addition, if Point \( P \) has a class label, but no point set that meets \( \epsilon - \text{nbhd}(p) \) cannot be found from the point, Point \( p \) will be classified as a boundary point. Therefore, several data sets can be obtained after every round of density clustering is executed, and each point can be classified as a class, boundary point or noise point.

This article proposes the trajectory identification method combined with density clustering. Compared with traditional tracking method, the method proposed in this article becomes a process of target identification, which can be used to identify the path of mesoscale eddies under global data, while turning the series system of the traditional algorithm into a way that supports batch processing of identification of multiple targets. By adding time series to the clustering process, we can avoid the conflict caused by the relatively far distance on the time scale while the closer distance on the space scale. The density clustering method regards each SLA data as a center point, selects its nearby points to join its cluster based on the distance. After several iterations, clusters are selected and the data set is divided by high-density areas and low-density areas. The area which mesoscale eddies passed could be regarded as high-density area, and the other space is regarded as low-density area.

The algorithm used in this article needs to define the parameters of the clustering process. The premise of the traditional tracking process is that the motion velocity of the mesoscale eddy is about 10 km/d. The eddy shall be deemed to have disappeared if it disappears for more than 3 days, and we will extend the search cycle to 5 days in this article. Therefore, the distance between data points with the time difference of more than 5 days is set as infinite in the definition of time scale, so as to avoid clustering error. According to the kinematic velocity and search cycle, it is necessary to search within the range of about 50 km around, and the eddy is deemed to have disappeared if no similar eddy is found. Under normal circumstances, the range of 1° on a map can be roughly represented as the distance of 120 km in reality. Therefore, the algorithm used in this article selects the coefficient search radius epsilon as 0.5, which represents that the search range is 0.5°, and the distance is about 60 km, slightly greater than the distance range of 50 km. The setting of another coefficient MinPts of density clustering requires sensitivity test, and the final value adopted in this article MinPts is 3. The experimental results are shown below:

Figure 1 shows the effect picture of two complex situations in two clustering processes, Figure 1(a,c) show the distribution of data points before clustering, of which Figure 1(a) shows the situation where an eddy disappears in the middle of the identification process, and if the eddy disappears for several days, it means that the originally dense points become sparse and then become dense. In the case of density clustering in this article, the red path refers to the path that ignores the situation of eddy disappearance as shown in Figure 1(b), and two parts are collectively referred to as one path, which is not disconnected due to the disappearance of the eddy. Figure 1(c) shows the disorderly distribution of data points in the area before clustering effect. If there is no discrimination of time scale, the path in the region will not be well differentiated. Figure 1(d) shows the situation where several paths are divided on time scale after the clustering process, and the lines of different colors represent different paths. As shown in Figure 1, the algorithm in this article can be used to differentiate the path regions using the method of model identification, which is more complete than the traditional algorithm. In the algorithm process, since the algorithm
is used to analyze data similarity from the angle of whole data, the process of algorithm processing can be turned into a process of batch processing, which can be used to search for subsequent nodes without the need to rely on the establishment of precedent noses in traditional algorithm.

Through clustering analysis, the motion point set of several mesoscale eddies can be obtained. The point set can be classified according to the class label of the point set, and then the basic motion trajectory of the mesoscale eddies can be obtained after ranking in the point set based on the time scale. By using clustering method, the batch processing could be added to the data processing. Firstly, the motion point is not based on the precursor-point, so it is possible to divide the whole data set to several fixed area. Secondly, the points are marked by class labels, it is more convenience to expand the range and merge with another trajectory. With the “Batch” processing, the operating speed could decrease exponentially.

However, there is still a number of data points with time conflict in some point sets among those point sets, which is manifested as two data points in the same day. In the trajectory, it is manifested as trajectory chaos or reentry. After that, the trajectory needs to be smoothed and noise points shall be eliminated to obtain the correct path.

**Filtering process**

The reason for the path set with time conflict is that one or several points that are close to the path location within the path range and the time scale is within 5 days. The possibility of such situation is that the mesoscale eddy is divided or a mesoscale eddy is formed nearby with smaller space and time scales, and the points in these two situations shall be considered as noise points for elimination.

Kalman filtering (Kalman, 1960) is a method to solve the current filtering problem of discrete data by recursive method. Its applied range includes multi-target tracking (Park & Lee, 2001), pure target location (Aidala, 1979) and image tracking (Weng et al., 2006). There are two main steps in the algorithm: 1) Use the tracking method to estimate the location of the next moment. 2) Select the results closest to the estimated results as subsequent nodes. The core of the algorithm is to use the probabilistic method to estimate the subsequent status through the status of precedent moments.
Coefficient initialization process of Kalman filtering process is shown below:

\[ P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

As the initial probability matrix, \( P \) is used to store the prediction probability in the computational process

\[ A = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

As the transfer matrix, \( A \) is applied to status switching

\[ Q = \begin{bmatrix} dt & 0 & 0 & 0 \\ 0 & dt & 0 & 0 \\ 0 & 0 & dt & 0 \\ 0 & 0 & 0 & dt \end{bmatrix} \sigma_p^2 \]

Covariance matrix \( Q \) to predict noise: Assuming that a Gaussian noise is superimposed on the prediction process, and the covariance matrix is \( Q \), and the variance \( \sigma_p^2 \) depends on the degree of trust of the prediction process. Assuming that the velocity of a moving object may not be uniform, the value of this diagonal matrix can be increased. If you want the trajectory to be smoother, you can lower the coefficient.

\[ H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \]

\( H \) represents the observation matrix

\[ R = \begin{bmatrix} \sigma_m^2 & 0 \\ 0 & \sigma_m^2 \end{bmatrix} \]

R represents the covariance matrix to observe noise, and parameter setting depends on the degree of trust of the observation process.

The process of Kalman filtering is divided into two steps. According to the following five formulas, the position of next moment can be calculated according to probability. According to the prediction, the noise points can be deleted and a smooth curve can be obtained. The time update step can be used to predict the location of next moment, and the main function of the measurement update step is to correct the error value combined with the actual value based on the predicted value, thereby eliminating the noise points.

Time update (prediction):

(1) Reckon the status variable of next moment:

\[ \bar{X}_{t+1} = AX_t \]

(2) Reckon error covariance of next moment:

\[ P_k = AP_{k-1}A^T + Q \]

Measurement updates (correction):

(1) Calculate Kalman gain:

\[ K_{t+1} = P_{k+1}H^T(HP_{k+1}H^T + R)^{-1} \]

(2) Update estimate based on observation variable \( Z_{t+1} \):

\[ \bar{X}_{t+1} = \bar{X}_{t+1} + (Z_{t+1} - H\bar{X}_{t+1}) \]

(3) Update error covariance:

\[ P_{k+1} = P_{k+1} - K_{t+1}HP_{k+1} \]

In the filtering process of the path, the number of continuous days of path is obtained by subtracting the time conflict subset from the number of continuous days of the data set. Later, the data of each day are provided with time update and measurement update. At the subset with conflict on time scale, noise points are abandoned due to greater difference from the predicted value.

Through the clustering process in last step, most of the paths obtained are suitable, which can directly output the results. However, still a few paths are irregular due to unstable detection results of mesoscale eddy or noise points of parameter setting in the path. It is specifically manifested as the situation where two or more than two data points occur in the path someday, and the situation leads to path reentry and large-angle turn, so the filtering method shall be applied to the noise removal process.

Figure 2 shows the situation of path conflict. The red dotted line represents the effect picture of path before the filtering process. For these detected trajectories, the path tends to be irregular with a reentry and a large-angle turn. The effect of filtering on the four paths, the path becomes smooth as shown in the blue curve in the figure, and the most obvious experimental effect is the effect with reentry phenomenon removed. The filtering effect can hardly coincide completely with the original path in detail, but the error range of the path filtering is controlled within 5%, considering the observation error, interpolation error and fusion error in the original data.

In terms of batch-processing capability, since batch processing targets at every data set with the same class label, the data set can be provided with batch processing to improve the filtering efficiency.

The parameter setting is based on data set attributes, For example, the range of 1° on a map can be roughly
represented as the distance of 120 km in reality. In this article, the motion velocity of the mesoscale eddy is about 10 km/d, 5 days disappear means the death of mesoscale eddy, that means the density clustering search radius is about 0.5°. The initialization of parameter MinPts should have a sensitivity test. As a whole, the value of MinPts for the spatial resolution 0.25° x 0.25° is lower than the MinPts for the spatial resolution 0.125° x 0.125°. Because Kalman filtering method has a good adaptable to all kinds of data set, the user could use the parameters proposed directly. These regulations are suitable for all kinds of data set.

**Experimental result**

Based on the above algorithm, the experiment in this article conducts a tracking test on the mesoscale eddies from January 1, 2010 to January 1, 2016 provided by AVISO, and selects the South China Sea (0°N–25°N, 105°E–125°E) as the experimental area. In this area, cyclonic and anticyclonic eddies are abundant and balanced, which is conductive to the evaluation of the tracking effect of the algorithm.

To display the effect of the algorithm in this article more intuitively in the picture, we select the mesoscale eddy trajectory within the time quantum from January 1, 2014 to January 1, 2016 to display, so as to avoid the situation where we cannot intuitively observe the mesoscale eddy trajectory due to excessive number of trajectories.

From Figure 3, we can see that the trajectory obtained is intricate, where blue curve represents the movement trajectory of cyclonic eddy and the red curve represents the movement trajectory of anticyclonic eddy. Due to the limitations of threshold setting, a few longer trajectories are cut off because of the longer disappearance time, but the situation is less than 2%. In the case of the remaining 98%, the algorithm in this article can accurately find the trajectory curve of the mesoscale eddies.

Figure 3 displays the eddy movement in 2 years. To better compare the algorithm performance, this article tests the algorithm using the method of lengthening the time of trajectory identification. The data set information is shown in Table 1, from which it can be seen that the size of data set scales up as time grows.
As the data set expands, the operation efficiency of the algorithm in this article, which is considered as a fast algorithm, is obviously higher than that of the traditional algorithm, whose running time is obviously shorter than that of the traditional algorithm as the data set enlarges.

In Figure 4, the left figure represents the identification time of cyclonic eddy trajectory while the right figure represents the identification time of anticyclonic eddy, the blue line represents the identification time of traditional algorithm while the red line represents the operation time of the algorithm in this article.

In terms of time comparison, the algorithm used in this article is shorter than that of the traditional algorithm. Firstly, the time complexity of the proposed algorithm is $O(n^2)$, it is lower than the traditional algorithm’s $O(n^3)$; Secondly, the proposed algorithm read all the data once, there is no need to reopen any

| Year   | Number of cyclonic eddies | Number of anticyclonic eddy |
|--------|----------------------------|-----------------------------|
| 2014–2016 | 3790                       | 4025                        |
| 2013–2016 | 5847                       | 6135                        |
| 2012–2016 | 7980                       | 8376                        |
| 2011–2016 | 10,290                     | 10,503                      |
| 2010–2016 | 12,305                     | 12,554                      |

Figure 3. Display of trajectory tracking of mesoscale eddies from 2014 to 2016 in South China Sea, the left picture is the standard path provided by AVISO while the right picture is the path obtained using the algorithm in this article.

Table 1. Number of data samples in different time quanta.
dossier. Since the results published by AVISO are based on the results of traditional algorithm published with a large number of manual interventions, the purpose of the algorithm in this article is to maintain a higher coincidence rate with the standard data set published by AVISO in results.

As shown in Table 2, the number of trajectories found using the algorithm in this article is greater or less than the standard. The reason is that some trajectories are incorporated into other trajectories that lead to a decrease in the number of trajectories, or the longer trajectory in the standard trajectories is interrupted for a longer time in the “jumping” process and the algorithm divides it into two trajectories, leading to an excessive number of trajectories. However, the difference in the number of trajectories found is controlled within 5% of the total number.

In terms of eddy comparison, this article takes the initial and end positions of mesoscale eddies as feature points to compare with the standard trajectory provided by AVISO, and the eddies found will be considered as correct eddies when the initial and end positions are the same.

$$\text{precision} = \frac{\sum (T_{\text{start}} = T_{\text{aviso-start}} \& T_{\text{end}} = T_{\text{aviso-end}})}{\text{Num}(T_{\text{aviso}})}$$  \hspace{1cm} (13)

Where, the numerator is the number of correct trajectories plus one when the initial and end positions are the same. Denominator is the number of trajectories provided by AVISO. Since the number of cyclonic eddy trajectories is nearly equivalent to that of anticyclonic eddy trajectories, the comprehensive accuracy rate takes the mean value of the accuracy rate of cyclonic and anticyclonic eddies, and the results obtained are shown in Table 3:

$$\text{precision}_{\text{mean}} = \frac{\text{precision}_{\text{CE}} + \text{precision}_{\text{AE}}}{2}$$  \hspace{1cm} (14)

The statistical structure in Table 3 shows that the accuracy of the algorithm used in this article in identification exceeds 90% compared with the coincidence rate of standard data set, that is to say, the algorithm in this article controls the error rate within 10%.

Table 2. Display of number of trajectories found in different time quanta.

|         | Standard trajectory of AVISO | Trajectory of the algorithm in this article |
|---------|------------------------------|--------------------------------------------|
|         | Number of cyclonic eddies    | Number of anticyclonic eddies              |
| 2014–2016 | 95                           | 90                                         |
| 2013–2016 | 138                          | 128                                        |
| 2012–2016 | 169                          | 182                                        |
| 2011–2016 | 231                          | 209                                        |
| 2010–2016 | 253                          | 275                                        |

Table 3. Comparison of accuracy rate under the data set in different time quanta.

|         | Accuracy rate of cyclonic eddy trajectories (%) | Accuracy rate of anticyclonic eddy trajectories (%) | Comprehensive accuracy rate (%) |
|---------|------------------------------------------------|--------------------------------------------------|--------------------------------|
| 2014–2016 | 92.89                                         | 96.84                                            | 94.865                        |
| 2013–2016 | 88.65                                         | 97.66                                            | 93.155                        |
| 2012–2016 | 91.81                                         | 90.70                                            | 91.235                        |
| 2011–2016 | 90.27                                         | 93.04                                            | 91.655                        |
| 2010–2016 | 91.81                                         | 92.60                                            | 92.2                           |

Conclusion

With regard to the tracking issue after mesoscale eddies are extracted from SLA data, this article proposes a tracking algorithm that supports batch process of mesoscale eddies based on density clustering on time scale and combined with Kalman filtering method. Different from the traditional tracking method that establishes the chain structure based on traversal and searches for subsequent nodes similar to precedent nodes to form a path, the algorithm proposed in this article turns the process of serial chain structure establishment into a process of target identification. Through the conversion of identification model, we search for mesoscale eddies with similar attribute among all mesoscale eddies under the limits of time and space scales and classify them into a class, so that the operation process of the algorithm in this article is no longer dependent on the establishment of precedent nodes. For the noise problem in the identification process, this article uses Kalman filtering to filter to ensure the purity of results.

With regard to “disappearance” and “jumping” problems in the life cycle of eddies, this article first expands the search scope and complements the disappearance process to solve the said problems by setting the threshold based on clustering. Then, it turns the sole series system in the traditional algorithm into a model identification way that supports batch processing. Finally, it proves that the method proposed in this article can greatly reduce the time complexity of the algorithm through the contrast experiment between traditional method and the method proposed in this article, and improves the operation efficiency of the algorithm with accuracy rate ensured.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Aidala, V.J. (1979, January). Kalman filter behavior in bearings-only tracking applications. IEEE Transactions on Aerospace & Electronic Systems, AES-15(1), 29–39. https://doi.org/10.1109/TAES.1979.308793
Birant, D., & Kut, A. (2007, January). ST-DBSCAN: An algorithm for clustering spatial-temporal data. Data & Knowledge Engineering, 60(1), 208–221. https://doi.org/10.1016/j.datak.2006.01.013
Bradley, P.S., Gehrke, J., & Ramakrishnan, R. (2002). Scaling mining algorithms to large databases. *Communications Of The Acm, 45*(8), 38–43. https://www.researchgate.net/publication/220420036_Scaling_mining_algorithms_to_large_databases

Chaigneau, A., Eldin, G., & Dewitte, B. (2009, July). Eddy activity in the four major upwelling systems from satellite altimetry (1992–2007). *Progress in Oceanography, 83*(1–4), 117–123. https://doi.org/10.1016/j.pocean.2009.07.012

Chaigneau, A., Gizolme, A., & Grados, C. (2008, October). Mesoscale eddies off Peru in altimeter records: Identification algorithms and eddy spatio-temporal patterns. *Progress in Oceanography, 79*(2–4), 106–119. https://doi.org/10.1016/j.pocean.2008.10.013

Chelton, D.B., Schlax, M.G., & Samelson, R.M. (2011, January). Global observations of nonlinear mesoscale eddies. *Progress in Oceanography, 91*(2), 167–216. https://doi.org/10.1016/j.pocean.2011.01.002

Doglioli, A.M., Blanke, B., & Speich, S. (2007, October). Tracking coherent structures in a regional ocean model with wavelet analysis: Application to Cape Basin eddies. *Journal of Geophysical Research Oceans, 112*(C05043). doi: 10.1029/2006JC003952

Ester, M. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Knowledge Discovery and Data Mining*, pp. 226–231.

He, Y., Tan, H., & Luo, W. (2012). MR-DBSCAN: An efficient parallel density-based clustering algorithm using MapReduce. *2011 IEEE 17th International Conference on Parallel and Distributed Systems* (pp. 473–480). IEEE.

Kalman, R.E. (1960, March). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering Transactions, 82*(1), 35–45. https://doi.org/10.1115/1.3662552

Manavalan, R., & Thangavel, K. (2011). TRUS image segmentation using morphological operators and DBSCAN clustering. *Information and Communication Technologies (WICT), 2011 World Congress on* (pp. 898–903). IEEE.

Nencioli, F. (2010, March). A vector geometry–based eddy detection algorithm and its application to a high-resolution numerical model product and high-frequency radar surface velocities in the Southern California Bight. *Journal of Atmospheric and Oceanic Technology, 27*(3), 564–579. https://doi.org/10.1175/2009JTECHO725.1

Okubo, A. (1970, June). Horizontal dispersion of floatable particles in the vicinity of velocity singularities such as convergences. *Deep Sea Research and Oceanographic abstracts, 17*(3), 445–454. https://doi.org/10.1016/0011-7471(70)90059-8

Park, S.T., & Lee, J.G. (2001, February). Improved Kalman filter design for three-dimensional radar tracking. *IEEE Transactions on Aerospace and Electronic Systems, 37*(2), 727–739. https://doi.org/10.1109/7.937485

Sun, M., Tian, F., & Liu, Y. (2017, February). An improved automatic algorithm for global eddy tracking using satellite altimeter data. *Remote Sensing, 9*(3), 206–224. https://doi.org/10.3390/rs9030206

Weng, S.K., Kuo, C.M., & Tu, S.K. (2006, June). Video object tracking using adaptive Kalman filter. *Journal of Visual Communication & Image Representation, 17*(6), 1190–1208. https://doi.org/10.1016/j.jvcir.2006.03.004

Yi, J., Du, Y., & Wang, D. (2017, November). Tracking the evolution processes and behaviors of mesoscale eddies in the South China Sea: Aglobal nearest neighbor filter approach. *Acta Oceanologica Sinica, 36*(11), 27–37. https://doi.org/10.1007/s13131-017-1136-6