18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Vessel Trajectory Partitioning Based on Hierarchical Fusion of Position Data

Xianbin Wu[1,2], Lin Wu[1,2], Yongjun Xu[1], Zhulin An[1], Boyu Diao[1,2]

[1]Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
{wuxianbin, wulin, xyj, anzhulin, diaoboyu2012}@ict.ac.cn

[2]University of Chinese Academy of Sciences
Beijing, China

Abstract – Maritime situational awareness now greatly benefits from AIS (Automatic Identification System). The most fundamental function a maritime situational awareness system should provide is vessel trajectory analysis. Previous work has shown the necessity of vessel trajectory partitioning before further analysis, but limited work existed for this problem. With billions of AIS messages received every month, computational complexity should be concerned as well as preciseness, and there is room for both of them to improve. This paper proposes a vessel trajectory partitioning method based on hierarchical fusion of position data reported by AIS, aiming at improving preciseness and speed. Our method consists of two levels: position fusion and sub-trajectory fusion. Experimental results based on real data demonstrate that our method splits routes and identifies stops precisely with the computational complexity of O(n).

Keywords: Maritime Situation Awareness, Automatic Identification System, AIS, trajectory partitioning, stop discovery, fuzzy logic.

1 Introduction

Ocean covers approximately 71% of the planet's surface, and most international trade is carried by it. In November 2014, China's president Xi Jinping announced plans to create a 40 billion USD development fund to help finance China's plans to develop the New Silk Road and the Maritime Silk Road. The prosperity on the ocean urges the analysis and surveillance of it. The main source of information for maritime situational awareness is currently AIS [1]. It is a self-reporting system installed on ships used to exchange information with other nearby ships, base stations and satellites. The information exchanged includes position, heading, rate of turn, navigation status, Maritime Mobile Service Identity (MMSI) number, etc. Ships of 300 gross tons and upwards on international voyages, 500 tons and upwards for cargos not in international waters and passenger vessels are required to fit an AIS transceiver [2].

The most fundamental function a maritime situational awareness system should provide is vessel trajectory analysis. It is the basis of vessel's behavior analysis, traffic pattern discovery, anomaly detection, etc. The advantages of trajectory partitioning have been explained by Jae-Gil [3], Nicolas [4], et al., but existing methods are not good enough in terms of preciseness and computation speed. We propose a vessel partitioning method based on hierarchical fusion of positions reported by AIS in this paper, which consists of two levels. In the first level, it aggregates position points of a vessel into sub-trajectories represented by line segments. In this way, original data is compressed to a much smaller volume without losing important information. Besides, compression speeds up the computation of higher level processes [5]. In the second level, sub-trajectories are fused into route segments and stops. Within a single route segment, a vessel is sailing in an almost constant direction. When a vessel is stopped, it may be in a port, anchoring area, or fishing area, etc. Those stops are important points of interest in maritime situational awareness. The computation complexity of our method is *O(n)* and the preciseness is validated by comparing results with nautical charts and Google Maps. Besides, for every level we go up in the hierarchy, the magnitude of *n* will decrease 1 or 2 orders.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 describes our methods. In section 4, we apply our method on real data set and compare with some other approaches. Conclusions and future works are summarized in section 5.

2 Related work

Giuliana et al. from NATO Science and Technology Organization (STO) Centre for Maritime Research and Experimentation (CMRE) presented a tool called TREAD (Traffic Route Extraction and Anomaly Detection) [6] [7] [8][9]. In their algorithms, routes are lines between two waypoints (stationary object, entry points and exti points, etc.) and clustering is performed on them. Jae-Gil et al. [3] observed that "clustering trajectories as a whole could not detect similar portions of trajectories". They proposed a framework called "partition-and-group", which portioned a trajectory into a set of line segments and then group similar line segments. Nicolas et al. illustrated the main advantages of trajectory partitioning with two examples [4].

There are two basic problems in partitioning vessel trajectories: trajectory segmentation and stop detection. Jae-Gil et al. [3] proposed a segmentation method based on the minimum description length (MDL) principle. As the cost of finding the optimal partitioning is "prohibitive", they devised an approximate solution, regarding the set of local optima as the global optimum. This approximate had a precision of about 80% on average, and authors proved that the time complexity is $O(n)$, because it required $N-1$ times of MDL computations. But when applying their method to partition vessel trajectories, the time complexity will be $O(n^2)$, as vessel' trajectories are straight lines most of the time and the time complexity of each MDL computation is $O(n)$. Gerben et al. adopted the Piecewise Linear Segmentation (PLS) to partition trajectories into linear segments by recursively keeping the points that have maximum error higher than a fixed threshold [5]. In this way, the maximum deviation was kept within the threshold. The worst case running time of PLS is $O(n^2)$. To reduce complexity, Nicolas et al. [4] adopted a simple Sliding Window algorithm which iteratively checks the distance between a point of the trajectory and its orthogonal projection on the straight line defined by the two previous observations. If the distance exceeds a threshold, the trajectory is split at the previous observation. Simple as it is, this algorithm will fail when the update interval of AIS messages is unequal or small (e.g., less than one hour).

Stops were usually detected by speed gating [5][7][9] [10]. But this may cause false negative because vessels will move even when anchored or moored due to the errors of GPS and timestamp, force of winds and waves, etc. Apart from this simple approach, there are four different methods developed to detect stops: IB-SMoT (Intersection Based Stops and Moves of Trajectories) [11], CB-SMoT (Clustering-Based Stops and Moves of Trajectories) [12] [4], DB-SMoT (Direction-Based Stop and Moves of Trajectories) [13], and a combination of CB-SMoT and DB-SMoT [14]. The IB-SMoT detects stops by finding intersection of trajectories with geographic objects. The CB-SMoT is a speed-based method, clustering the slowest sections of a trajectory. The DB-SMoT is a clustering method based on the direction variation. Fabio et al. [14] combined CB-SMoT and DB-SMoT to discover fishing stops.

3 Vessel Trajectory Partitioning

This section describes our method of vessel trajectory partitioning, see Figure 1.

Our idea is somewhat similar to that of [15], which used hierarchical concepts to describe motions. They "built up increasingly abstract concepts describing qualitative features of motion on top of lower level concepts which in turn depend at the lowest level on the concrete observations provided by the raw data". The key idea of our method is to describe trajectories by hierarchical concepts. In the first level, the partitioning of trajectories is in nature grouping positions in raw AIS records into sub-trajectories represented as straight line segments. In the second level, we aggregate certain sub-trajectories into more abstract concepts: route segments and stops.

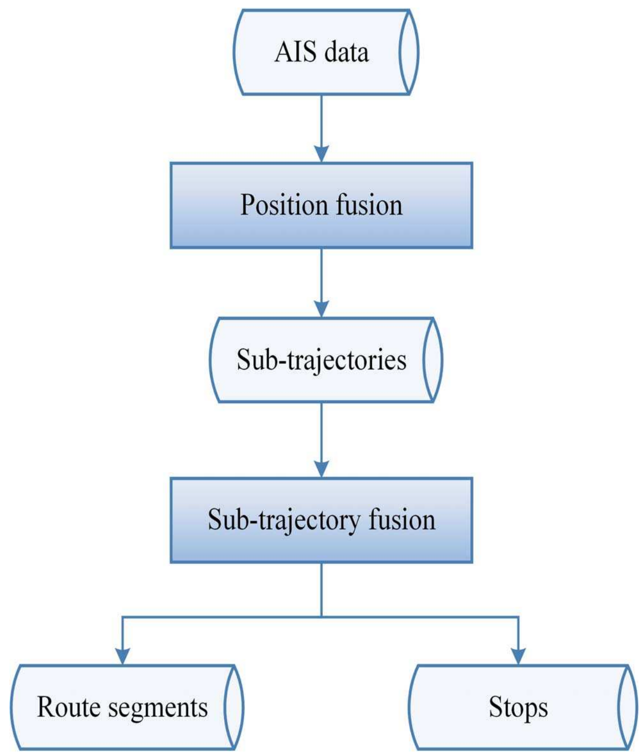

Figure 1. Flowchart of trajectory analysis. Our hierarchical vessel trajectory partitioning method is in the blue dotted box.

Let Tr_{id} be the trajectory of vessel *id* where *n* is the total number of AIS reports.

$$Tr_{id} = \{P_1, P_2, \cdots, P_n\} \tag{1}$$

In the above equation, $P_i = (longitude_i, latitude_i)$ is the position of the ship from GNSS fixed on ship. We work with only position data, because other data like Speed Over Ground (SOG) and Course Over Ground (COG) are not reliable.

3.1 Position fusion

In level 1 process, we aggregate consecutive positions of each vessel into several sub-trajectories so that the course is almost constant within a single sub-trajectory $subTr_{id,k}$.

$$Tr_{id} = \{subTr_{id,1}, subTr_{id,2}, \cdots, subTr_{id,m}\} \tag{2}$$

$$subTr_{id,k} = \{P_i, P_{i+1}, \cdots, P_n \mid \forall 1 \le i < j < n,$$
$$\left| Angle(\overrightarrow{P_i P_{i+1}}, \overrightarrow{P_j P_{j+1}}) \right| \le \alpha_{sub-tr} \} \quad (3)$$

The threshold α_{sub-tr} is set as 30° in this paper. As we know, the angle of vectors can be represented as formula (4).

$$\left| Angle(\overrightarrow{P_i P_{i+1}}, \overrightarrow{P_j P_{j+1}}) \right| =$$
$$\left| Angle(\overrightarrow{P_1 P_2}, \overrightarrow{P_j P_{j+1}}) - Angle(\overrightarrow{P_1 P_2}, \overrightarrow{P_i P_{i+1}}) \right| \quad (4)$$

Thus, the condition of sub-trajectory partitioning is converted to formula (5).

$$\max(Angle(\overrightarrow{P_1 P_2}, \overrightarrow{P_j P_{j+1}})) - \min(Angle(\overrightarrow{P_1 P_2}, \overrightarrow{P_i P_{i+1}}))$$
$$\le \alpha_{sub-tr} \quad (5)$$

We compute the angle of every two consecutive points with the first two points in the sub-trajectory, and record the maximum (non-negative) and minimum (non-positive) one. When the difference between these two angles is higher than threshold, a new sub-trajectory starts. See Algorithm 1 and Algorithm 2 for details. The complexity is *O(n)*.

3.2 Sub-trajectory fusion

In sub-trajectory fusion, we aggregate sub-trajectories into route segments and stops. This step is also referred to as “semantic partitioning” in literature [4]. Within a single route segment, sub-trajectories are almost in the same direction. Stops are formed when ships are moored in ports or anchored in anchoring areas. We first aggregate sub-trajectories into stops. Then we aggregate other sub-trajectories into route segments. The input of this process is about 2 orders less than that of position fusion, and the output is reduced by 1 to 2 orders further.

To detect stops, we observed the patterns of vessels’ sub-trajectories first. The result of our observation is depicted in Figure 2, which is based on 30000 AIS messages. From this figure, we can see that the length of sub-trajectories is generally under 30 meters when a vessel is moored or anchored. And majority of angles are above 60°. According to these patterns, we can estimate the degree of membership of each sub-trajectory to the state stopped, see Figure 3.

Algorithm 1 Position fusion

```
Require: Tr // The whole trajectory, which is made of positions in raw AIS records
Require: CurrentSubTr // The subjectory which is now looking for its end point
Output: ListSubTrs //List of sub-trajectories, each of them consists of consecutive points
For all position in Tr:
    If IsNewSubTr(CurrentSubTr, position): // If position is the beginning of a new sub-trajectory
        ListSubTrs.append(CurrentSubTr)
        CurrentSubTr.clear()
        CurrentSubTr.append(position)
    Else: // If position is NOT the beginning of a new sub-trajectory
        CurrentSubTr.append(position)
    End If
End For
```

Algorithm 2 Splitting point detection : IsNewSubTr(CurrentSubTr, position)

```
Require: CurrentSubTr, position // The subjectory which is now looking for its end point
Require: MaxAngle, MinAngle // The maximum and minimum angle of every two consecutive points with the first line segment in CurrentSubTr
Require: ThresholdAngle //Threshold of the difference between MaxAngle and MinAngle, set as 30°
Require: position //The position which is now computing if it is the beginning of a new sub-trajectory
If length(CurrentSubTr) < 2:
    Return False
ComparingVector = Vector(CurrentSubTr[lastIndex], position) // The vector from CurrentSubTr[lastIndex] to position
ComparedVector = Vector(CurrentSubTr [0], CurrentSubTr [1]) // The vector from the first point to the second point
Angle = Angle(ComparingVector, ComparedVector)
UpdateMinMax(angle)
If MaxAngle–MinAngle > ThresholdAngle:
    Return True // The position is the beginning of a new sub-trajectory
Else:
    Return False // The position is not the beginning of a new sub-trajectory
```

To filter noises, we use an integral filter to integrate membership of consecutive sub-trajectories. But a problem arises when performing integration: the integrated membership will approach infinity as the degree of membership is non-negative. To solve this problem, we extend the range of traditional membership functions to negative values, see Figure 3. And the integrated degrees of membership are confined to within [0, 1]. When the integrated degree of membership reaches 1, a stop is detected. And the vessel starts moving again when the degree of membership reaches 0. For details, see Algorithm 3. The evolution of the integrated degree of membership is depicted in Figure 4 when a vessel entered and left a port.

After stops are detected, we aggregate other sub-trajectories into route segments, so that the maximum direction deviation within a single route segment is less than threshold. This process is similar to Algorithm 1 and Algorithm 2, except that atomic elements are sub-trajectories represented as line segments instead of position points.

4 Experimentation and Results

We applied our method on a data set containing 473963 AIS records from 10 vehicle carriers along the coast of China. Algorithms are implemented by python 2.7.3 on a notebook with Intel Core2 Duo CPU T6600 @2.20GHz, 2GB RAM. The average execution time of our two level partitioning on each vessel is around 2.30 seconds and 0.22 seconds respectively. As a reference, it takes about 0.1 seconds to perform the operation of adding two integers

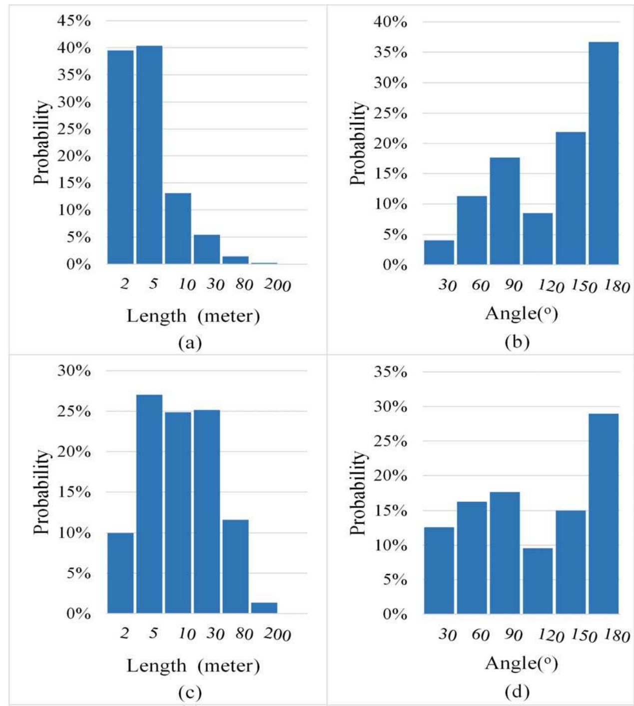

Figure 2. (a) Distribution of sub-trajectories' length when a vessel is moored in ports; (b) Distribution of angles between two consecutive sub-trajectories when a vessel is moored in ports; (c) Distribution of sub-trajectories' length when a vessel is anchored; (d) Distribution of angles between two consecutive sub-trajectories when a vessel is anchored.

Algorithm 3 Sub-trajectory fusion - Stop detection

```
Require: ListSubTrs //List of sub-trajectories from L1 Trajectory partitioning
Require: MinMembershipPotentialStop // The minimum membership for a sub-trajectory to be a potential stop, set as 0.2
Require: ListSubtrsBelongingToPotentialStop // List of consecutive sub-trajectories which belong to POTENTIAL_STOP
Output: ListStops // List of Stops, each of them consists of consecutive sub-trajectories
μ_STOP(ListSubtrsBelongingToPotentialStop) = 0
CurrentState = MOVE // The state of ListSubtrsBelongingToPotentialStop: MOVE, POTENTIAL_STOP, STOP
For all subTr_i in ListSubTrs:
    μ_STOP(ListSubtrsBelongingToPotentialStop)=max(0,min(1,μ_STOP(ListSubtrsBelongingToPotentialStop)+μ_STOP(subTr_i)))
    If μ_STOP(ListSubtrsBelongingToPotentialStop)= 1:
        CurrentState = STOP
        LastStopSubTr = subTr_i // record the last subtrajectory of Stop
        ListSubtrsBelongingToPotentialStop.append(subTr_i) // Add current sub-trajectory to the list
    Else if μ_STOP(ListSubtrsBelongingToPotentialStop) >= MinMembershipPotentialStop :
        If CurrentState=MOVE: //If it is MOVE before, then it is POTENTIAL_STOP
            CurrentState = POTENTIAL_STOP:
        ListSubtrsBelongingToPotentialStop.append(subTr_i) // If it is STOP before, then still in STOP state
    Else: //μ_STOP(ListSubtrsBelongingToPotentialStop)< MinMembershipPotentialStop
        If CurrentState is STOP: //The vessel leaves the state of STOP
            ListStops.append(ListSubtrsBelongingToPotentialStop(0, LastStopSubTr))
        ListSubtrsBelongingToPotentialStop.clear(); //The vessel is in the MOVE state
        CurrentState = MOVE
    End if
End for
```

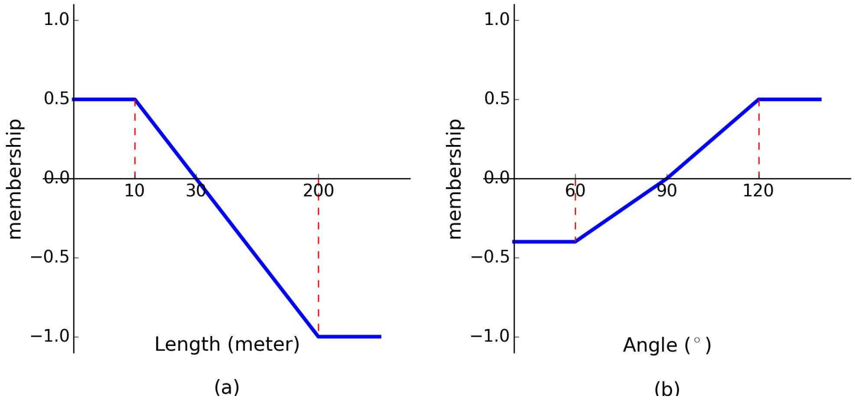

Figure 3. The membership function of a subjectory belonging to a stop according to its length and angle with respect to the previous subtrajectory respectively. The final membership is determined by their intersection.

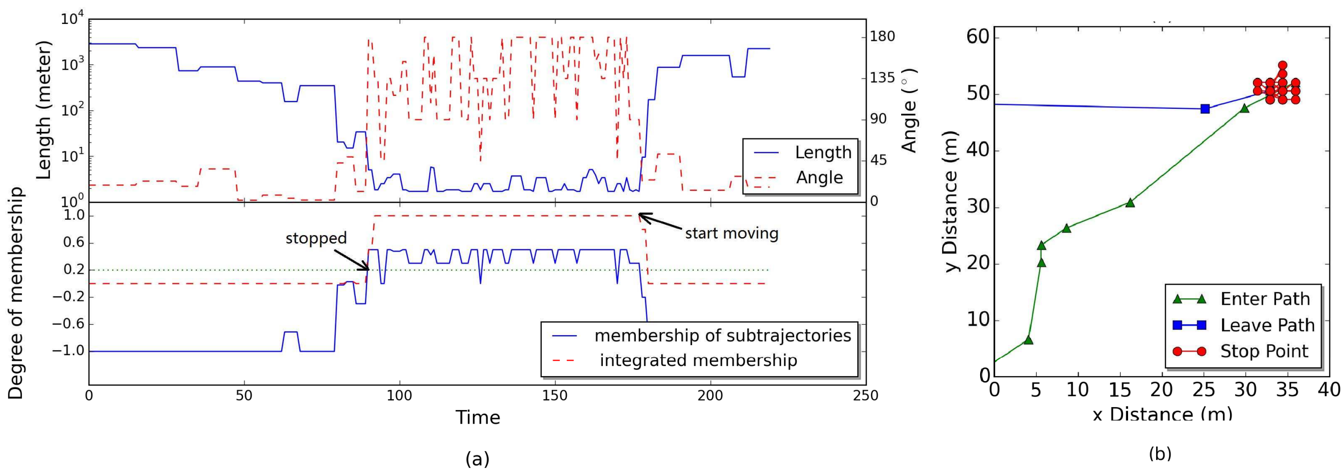

Figure 4. These figures are dipicted according to real data, corresponding to a vessel entered and left Dongguan, a port in Guangdong, China. The first figure on the left is the degree of membership that each sub-trajectory is in the state stopped according to length and angle respectively. The second figure on the left dipicts the evolution of the integrated degree of membership to stopped when a vessel entered and left a port.

600000 times. This proves the efficiency of our hierarchical framework, and the higher the level, the less time will be needed.

4.1 Position fusion

In September 2013, we received 32176 AIS messages from the vessel whose MMSI is 41320xxxx. Using our method, we aggregated them into 1297 sub-trajectories.

An example of the results of route segmentation is depicted in Figure 5. It is based on AIS records from a vessel sailing along the coast of China from 2nd Sept to 11 Sept 2013.

4.2 Sub-trajectory fusion

After sub-trajectory fusion, we got 250 route segments and 16 stops. A part of result of two level fusion is depicted in Figure 5. Route segments and stops are represented by red line segments and blue dots respectively.

We compare our method with speed gating and combination of CB-SMoT and DB-SMoT [14] in Figure 6. Our preciseness outperforms both of them.

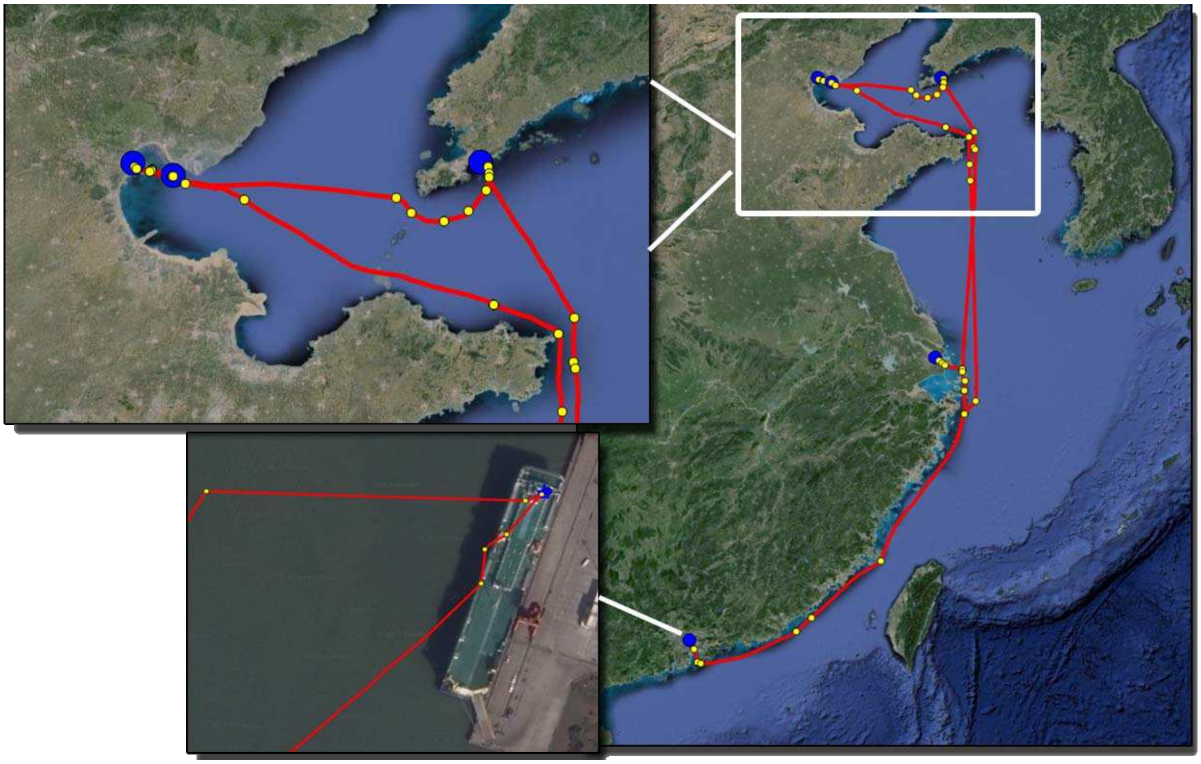

Figure 5. Result of trajectory partitioning of a vessel sailing along the coast of China from 2 Setp. to 11 Setp. 2013. The red Line segments are route segments, split by yellow circles. The blue dots represent the stops dectected.

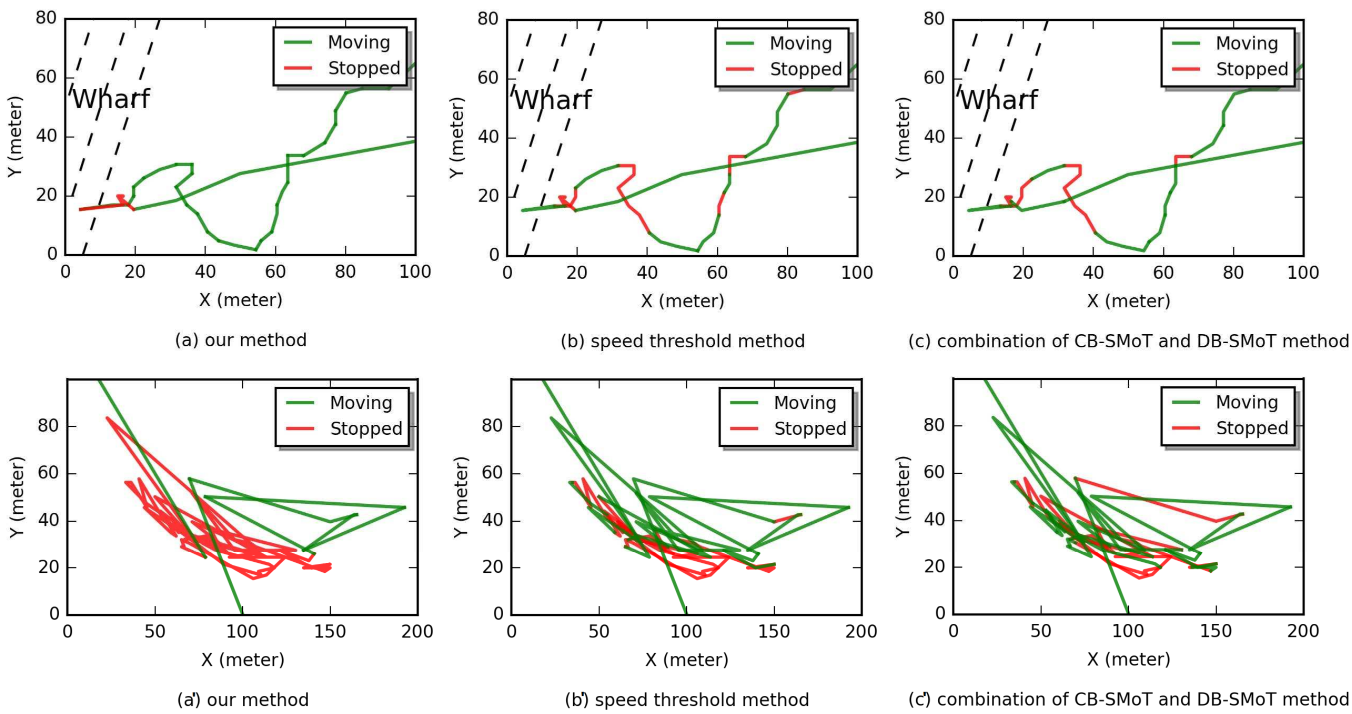

Figure 6. The results of stop detection by three methods when a vessel is anchored. The speed threshold used in the latter two methods is 0.3 knots. The COG variation threshold used in the third method [14] is 10°.

5 Conclusion and future work

Previous work has shown the necessity of vessel trajectory partitioning, but limited work existed for this problem. With billions of AIS messages received every month, computational complexity should be concerned as well as preciseness, and there is room for both of them to improve. This paper proposes a hierarchical vessel trajectory partitioning method, aiming at improving preciseness and speed. Our method consists of two levels: position fusion and sub-trajectory fusion. Experimental results demonstrate that our method splits routes and identifies stops precisely.

Our idea of trajectory partitioning is somewhat similar to that of [15], which used hierarchical concepts to describe motions. But we used fuzzy sets instead of hard threshold to define concept boundaries in the second level. Our future work would be to add higher levels into our framework. Stops would be further labeled as wharves, anchorages, fishing points, etc. And some route segments would be aggregated into turning circles, fishing route, noises due to ships floating freely, etc. After that, we plan to aggregate those entities extracted from different ships to find seaways, ports, fishing areas, etc.

As data volume of AIS messages is relatively huge, we will also implement our methods on distributed platforms like Spark.

6 Acknowledgement

This paper is supported by Important National Science & Technology Specific Projects under grant No.(2014ZX03006-003). And we would like to thank China Transport Telecommunications & Information Center for providing us the global AIS data.

References

[1] Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band, International Telecommunications Union (ITU) Recommendation ITU-RM.1371-4, April 2010.

[2] "Safety of life at sea (solas) convention chapter v, regulation 19."

[3] Lee, J.-G., Han, J., and Whang, K.-Y., "Trajectory clustering: a partition-and-group framework," ACM SIGMOD international conference on Management of data 2007, pp. 593-604.

[4] N. Le Guillarme and X. Lerouvreurt, "Unsupervised Extraction of Knowledge from S-AIS Data for Maritime Situational Awareness," 16th International Conference on Information Fusion, 2013, pp. 2025-2032.

[5] G. K. D. Vries and M. Someren, "Machine learning for vessel trajectories using compression, alignments and domain knowledge," Expert Systems with Applications, vol. 39, no. 18, pp. 13426-13439, 2012.

[6] G. Pallotta, M. Vespe and K. Bryant, "Traffic Route Extraction and Anomaly Detection from AIS Data," International COST MOVE Workshop on Moving Objects at Sea, Brest, France.

[7] G. Pallotta, M. Vespe and K. Bryant, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," Entropy, vol.15, no. 6, pp. 2218-2245, 2013.

[8] G. Pallotta, M. Vespe and K. Bryant, "Traffic knowledge discovery from ais data," 16th International Conference on Information Fusion, 2013, pp. 1996-2003.

[9] M. Vespe, I. Visentini, K. Bryan, et al, "Unsupervised learning of maritime traffic patterns for anomaly detection," 9th IET Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, pp. 1-5.

[10] R.O. Lane, D.A. Nevell, S.D. Hayward, et al., "Maritime anomaly detection and threat assessment," 13th International Conference on Information Fusion, 2010, pp. 1-8.

[11] L. O. Alvares, V. Bogorny, B. Kuijpers, et al., "A model for enriching trajectories with semantic geographical information," in Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems, New York, USA, 2007, pp. 22:1–22:8.

[12] A. T. Palma, V. Bogorny, B. Kuijpers, et al., "A clustering-based approach for discovering interesting places in trajectories," in Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08. New York, USA, 2008, pp. 863–868.

[13] J. Rocha, G. Oliveira, L. Alvares, et al., "Db-smot: A direction-based spatio-temporal clustering method," 5th IEEE International Conference on Intelligent Systems, July 2010, pp.114–119.

[14] F. Mazzarella, M. Vespe, D. Damalas, et al., "Discovering vessel activities at sea using AIS data: Mapping of fishing footprints," 16th International Conference on Information Fusion, 2014, pp. 1-7.

[15] V. Avram, U. Glasser, and H.Y. Shahir, "Anomaly detection in spatiotemporal data in the maritime domain,"

IEEE International Conference on Intelligence and Security Informatics (ISI), 2012, pp. 147-149.