

# Apprenticeship Learning of Ship Behavior in Crowded Area by Dimension Compression

Yuxin Liang and Masayoshi Mase

**Abstract**—There is a strong demand for autonomous ship navigation systems in maritime logistics. Such systems need to be able to forecast behaviors of other ships accurately to avoid collisions. Here, time-series of ship positions, called AIS data, can be used in apprenticeship learning (AL) by defining an object map created from the data as a state and the turning direction of the ship as an action. However, when we analyzed 1 months' worth of AIS data, none of the generated path data took actions in the same state pattern twice. This paper proposes to use a Co-Moving Frame (CMF), a local segment of the environment on a small timescale. CMF improved the effectiveness of the data usage, and as a result, AL forecast paths of ships with 81.2% accuracy when applying CMF. This result is 29.2% better than that of a state transition model generated from the same dataset without applying CMF.

**Index Terms**—Apprenticeship Learning, Neural Network, Q-learning, Ship, AIS

## I. INTRODUCTION

There is trend in logistics to substitute human labor by artificial intelligence (AI). In maritime logistics, in particular, autonomous navigation of ships in various environments, ranging from crowded harbor areas to wide ocean areas [1], is seen as a way of reducing labor costs.

The primary requirement of the navigation operation is to avoid collisions with surrounding ships. Therefore, it is necessary to forecast the behavior of surrounding ships in various situations for planning a safe path.

The behavior of surrounding ships can be forecasted by using agent simulation incorporating behavior rules. Such behavior rules can be summarized from datasets consisting of path records of ships. Here, global open data, i.e., Automatic Identification System (AIS) data [2], are available for extracting time-series of ship positions.

The objective of this study is to forecast the behavior of ships in various situations by using time-series position data. Various approaches to forecasting a ship's behavior have been studied. In particular, Simsir et al. proposed to use an artificial neural network based decision support system to avoid ship collisions [3]. The method forecasts positions of multiple ships three minutes ahead. However, the system requires a predefined reference route for the environment and the accuracy of forecasting decreases when the ship is far from the route.

Zhang *et al.* proposed to use a Bayesian network for ship navigation support [4]. The method analyzes parameters in

behavior rules based on the distance between and directions of two ships. It was shown to be applicable to large ocean areas with few ships nearby, but was not tested in more crowded environments with more than two ships.

Gorge et al. proposed a grid-based statistical analysis of a ship's behavior from a dataset of ship paths [5]. This method is applicable for analyzing crowded areas. However, the grid has limited resolution due to computational complexity, and the method is difficult to apply in an ocean environment.

Shirai et al. analyzed differences in ship behavior in Tokyo Bay by categorizing the sizes of ships and constructing a traffic flow network [6]. However, the model they created is not useful for real-time forecasting.

All of the existing approaches to forecasting the behavior of ships have difficulties. Therefore, it is reasonable to look for an approach from a different domain as a way to solve the problem.

Abbeel *et al.* proposed apprenticeship learning (AL) for imitating experts' behavior by statistically analyzing datasets of expert actions in certain situations [7].

Sharifzadeh et al. extended AL to enable it to analyze driving behavior in a highway simulator [8]. Their method forms behavior rules with which car can avoid collisions in the simulator. It uses a Markov decision process (MDP) [9] based agent model. The MDP model is applicable to both crowded and wide open areas.

We consider that application of AL by defining an MDP for ships' behavior is a promising approach to achieving our goal. One typical way of designing an MDP based ship model is to make an object map from AIS data as a state and the ship's turn in a certain direction as an action.

However, such a method is not directly applicable to the maritime industry due to a difference in problem settings. A real bay area environment has too many possible patterns in any particular situation. For example, situations in a bay area may be classified into patterns of elements in the environment (called features), including distances and directions to land, the destination, and surrounding ships. In crowded environments, the numbers of combinations of features and corresponding situations are huge. Moreover, the quantity of AIS data available for analyzing real environments is too sparse. AL is applicable to datasets of path records fitting a certain situation, yet in a real environment, the same situation will likely never happen again. In addition, most of the grids are empty in a typical object map for a large environment. Therefore, using an object map is an inefficient way of expressing the situation.

This paper proposes data management using a data structure conversion called Co-Moving Frame (CMF). CMF translates a record of a situation into various datasets based on

Manuscript received November 30, 2017; revised April 1, 2018.

Yuxin Liang and Masayoshi Mase are with Hitachi, Ltd, Kokubunji, Tokyo 1858601 Japan (e-mail: yuxin.liang.na@hitachi.com, masayoshi.mase.mh@hitachi.na.com).

a partially observed situation from the viewpoint of the pilot or navigator of the ship. CMF conducts a local segmentation of the environment. By applying CMF, situational data can be converted into distances and directions to other features, such as surrounding ships, land, and the destination. By converting data into relative distances, the number of feature patterns can be limited. Thereby, CMF resolves the sparseness problem of the AIS data.

The contributions of this study are summarized as follows.

- A ship behavior model based on a Markov decision process was developed that is able to forecast paths of ships.
- A data management methodology was devised to solve the data sparseness problem of AIS data.
- An effective dataset for training the MDP agent model was created by apprenticeship learning.

## II. PREREQUISITES

MDP is used for defining the ship behavior model. AL and a Deep Q Network (DQN) [10] are used for the behavior analysis. Following is the terminology of this research.

### A. Markov Decision Process

In MDP, an agent takes an action:  $a$  for maximizing an accumulative reward:  $r$  in certain states:  $s$ . The action of the agent is decided on the basis of a policy:  $\pi(a|s)$  at each time step:  $t$ . Agents transit their state in accordance with a state transition probability:  $P(s'|s,a)$ . By their actions, agents obtain a reward based on a reward function:  $R(s,a)$ .

### B. Apprenticeship Learning

AL estimates the optimum policy  $\pi(a|s)$  from the state transition paths (namely episodes) of experts' actions by fitting  $\pi(a|s)$ . When a large number of states are available for the problem, each state is mapped into a lower dimension vector, called a feature vector  $\varphi(s)$ .  $\varphi(s)$  represents environmental conditions that may be components of the reward function. The estimated optimized reward function  $R(s)$  weights the feature vector  $\varphi(s)$  in accordance with the following equation, where  $\omega$  represents a vector of weights for the feature values.

$$R(s) = \omega \cdot \varphi(s) \quad (1)$$

AL uses the feature expectation value  $\mu$  to represent the accumulative feature value  $\varphi(s)$  of an episode in the learning process. It minimizes the gap between the expectation value from the expert's demonstration ( $\mu_E$ ) and the expectation value from the estimated reward function ( $\mu(\pi)$ ). These expectation values are calculated as follows.

$$\mu_E = \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{\infty} \gamma^t \varphi(s_t^{(i)}) \quad (2)$$

$$\mu(\pi) = E[\sum_{t=0}^{\infty} \gamma^t \varphi(s_t) | \pi] \quad (3)$$

$$\|\mu(\pi) - \mu_E\|_2 \leq \varepsilon \quad (4)$$

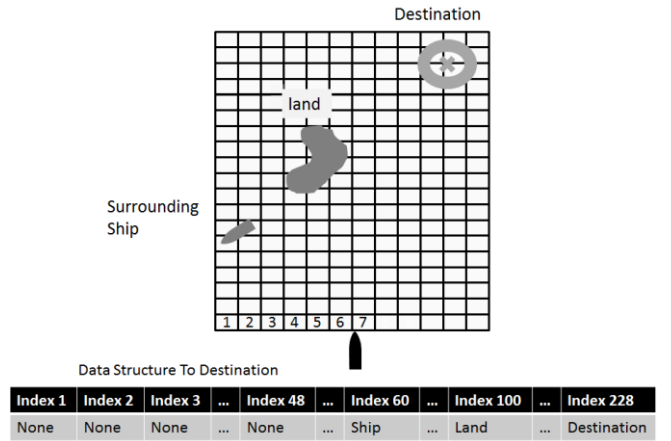


Fig. 1. Traditional sort of environment mapping.

### C. Reinforcement Learning and Deep Q Network

Reinforcement learning (RL) is a method to maximize the expectation value of cumulative  $R(s,a)$  (called  $Q(s,a)$ ) by estimating the optimum policy  $\pi(a|s)$  from an episode, or the series of records which represent states and actions of the agent. The DQN estimates the result of RL by using a neural network. The input of the network is a feature vector  $\varphi(s)$  of the environment, and the output is a  $Q(s,a)$  value for each available action. Due to its high accuracy and low computational complexity, DQN is widely used for solving problems in environments with many states.

## III. PROGRAM SETTING

We used AL for analyzing the behavior of ships in the waters of Tokyo Bay, Japan. We used an AIS dataset [11] collected from September 1st to September 30th, 2015 in sea areas between  $139^{\circ}37'4''$  -  $140^{\circ}06'0''$  East Longitude, and between  $35^{\circ}20'31''$  -  $35^{\circ}32'42''$  North Latitude (we call the region Area A). Area A is notorious for frequent collisions of ships [12]. To get to Area A from outside of the bay, all ships must go through the Uraga Channel [13], where ships are obliged to sail in formation. When ships arrive at Area A after passing through the channel, they are no longer obliged to form a line and can move freely towards their destination. The steering of ships in Area A becomes complicated because each ship must now avoid collision with other ships in the crowded environment. Because of the danger posed, local governments have demanded a means of forecasting the behaviors of ships in Area A.

Fig. 1 shows an image of the conventional MDP by mapping an environment onto a grid. Each shaded square in the figure represents a grid of the map. Each grid has a unique index and feature information. As shown in the Data Structure row in Fig. 1, the feature information can be either "None" or any defined feature. In this map, a state is a pattern of values stored in the grids, and an action is the transition through the grids of the ship being analyzed. This sort of map is widely used for behavior analysis of mobility because of its ease in

defining a model when something is able to move freely [14].

However, such a map is not effective for analyzing the behavior of ships moving in a large space. The key concept of AL is matching paths generated from the MDP with actual patterns observed in certain situations. If AL conducts path matching with an observation pattern summarized from only a few records of actual paths that ships took, the agent might learn from exceptional actions rather than common ones in that situation. Therefore, it is important to obtain a path dataset that contains multiple action records in the same state pattern for AL. When we analyzed AIS data, as discussed in Section 7, none of the generated path data took actions in the same state pattern twice. This was due to sparseness of data resulting from the huge number of possible combinations in the situation patterns.

By denoting the category of features as  $M$  and the number of grids for analysis as  $N$ , the total number of combinations of patterns in the environment can be expressed as

$$Total\_Availble\_Pattern = (M + 1)^N \quad (5)$$

When applying AL to ship's behavior analysis in Tokyo Bay Japan, we set  $M = 3$  and  $N = 3.9 * 10^6$  (the calculation is in the Appendix). The equation shows that the combinations of patterns highly depends on  $N$ . The total number of available state patterns is quite large. Therefore, the chances of collecting path data on the same situation again are quite small. Therefore, the data sparseness problem appears when AL is used to analyze AIS data in Area A.

#### IV. MDP-BASED SHIP MODEL

We solve the data sparseness problem by converting the data and defining an MDP based ship model in relative terms.

In particular, we propose a relative data format. As shown in the Data Structure row of Fig. 1, when analyzing a large environment with a few features, most of the grids in the map have the feature of "None". Since the information "None" is not so valuable in the analysis, we can compress the information into a simpler form by converting the features into relative distances and directions. Accordingly, the total number of available patterns becomes

$$Total\_Available\_Pattern = \prod_{i=0}^M Y^X \quad (6)$$

Here,  $X$  represents the categories of feature values and  $Y$  represents the types of value for each feature value.  $X$  and  $Y$  can be defined arbitrarily for the sake of balancing accuracy and computing complexity. Therefore, the total number of available patterns can be controlled regardless of the situation. This approach can alleviate the data sparseness problem. We call this data structure conversion Co-Moving Frame (CMF).

Fig. 2 shows the flow of AIS data in the behavior analysis applying CMF. A traditional environment map data is converted into CMF data. To generate the traditional environment map, data cleansing and data management are applied to the AIS data. The data cleansing part removes defective data. The data management part generates path data

by chronologically aligning the position data. After that, the environment dataset and path dataset are generated. The environment dataset contains information on features in the format of a grid map while the path dataset contains the grid transitions in the paths of ships. We use these datasets to generate the relative environment and action datasets of CMF.

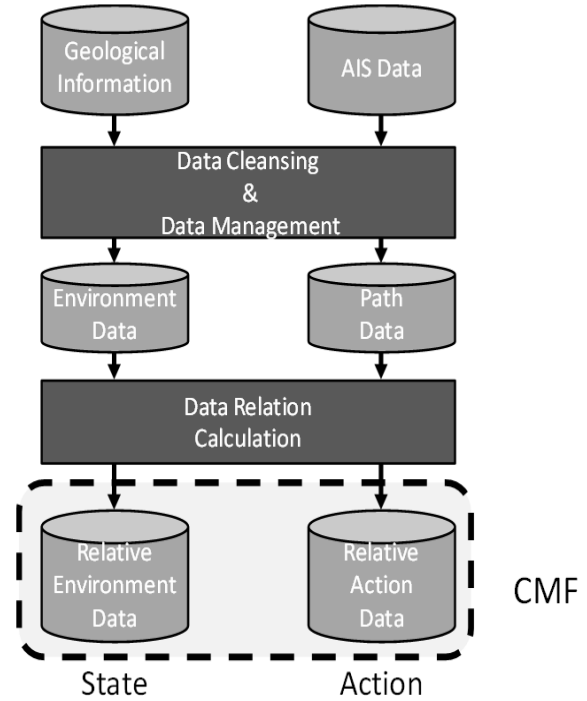


Fig. 2. Data flow for analyzing ship's behavior with CMF.

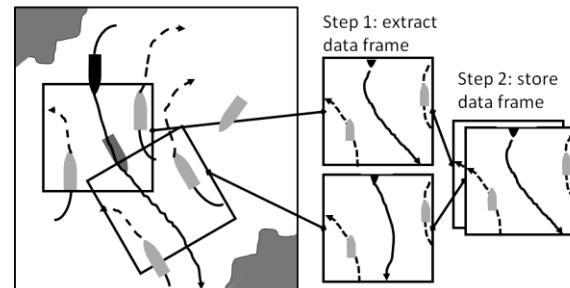


Fig. 3. Image of process of applying CMF to ship's environment.

In CMF, a state is defined as relative distances to features and an action is defined as a change in the status of motion of an agent, including rotations in heading.

Fig. 3 shows an image of the process to apply concept of CMF into a ship's driving navigation environment. In Step 1 of the process, a local extraction of environment is extracted from the entire environment. In Step 2, local extraction is conducted by converting environmental information into relative distances and directions as CMF data format. By applying CMF, data structures for reflect different situations can be communized. Therefore, the extracted CMF can be used for statistically analyzing the behaviors of ships.

#### V. DEFINITION OF CMF FOR AIS DATA

##### A. Data Cleansing and Data Management

The Automatic Identification System (AIS) [2] is a system for informing ships of their conditions by using electrical

signals. AIS data consist of records of decoded AIS signals picked up by the receiver. The data includes the following data fields:

- 1) Date and time when the information was sent
- 2) Position of ship
- 3) Direction of ship
- 4) Velocity of ship
- 5) Destination of ship
- 6) Maritime Mobile Service Identity (MMSI)

MMSI is the identification number of the AIS transmitter. The information is transmitted as electrical signals; occasionally some data are lost in a crowded area due to signal interference [15]. In addition, the destination information of AIS data typically contains many errors because it is inputted manually [16]. Thus, AIS data needs to be cleaned up before it is analyzed.

Algorithm 1 Data cleansing & Data management in AIS Data

- Inputs: AIS dataset  $S\_AIS$ , Geological Information  $S\_GEO$   
 Outputs: Environment Dataset  $S\_env$ , Path Dataset  $S\_path$
- 1: For each records in  $S\_AIS$ :
  - 2: Sort the records by date and MMSI and index them.
  - 3: Generate an episode from the records, store it as  $S\_path$ .
  - 4: For each records in  $S\_path$ :
  - 5: Expand date expression into YY:MM:DD:h:m:s.
  - 6: Make a grid environment by referring to Geohex
  - 7: Amend the destination information in AIS data based on the actual port that ship arrived at in the future records.
  - 8: Restore lost or missing data by interpolation.
  - 9: Use a Kalman filter to reduce the amount of noise in the ship position data.
  - 10: Amend the velocity information in accordance with the change in the ship's position during the episode.
  - 11: Append information on the future position
  - 12: Refresh the values in  $S\_path$ .
  - 13: Store records of ships' positions and geographical properties of the grid in  $S\_env$ .
  - 14: Export  $S\_env$  and  $S\_path$  datasets.

Fig. 4. Data cleansing and data management algorithm.

Fig. 4 shows the data cleansing and data management process. The input includes AIS datasets  $S\_AIS$  and a geographic information dataset  $S\_GEO$ . The output of the algorithm is the environment dataset  $S\_env$  and path dataset  $S\_path$ . Steps 1 through 3 apply the data management process to  $S\_AIS$  in order to generate  $S\_path$ . Steps 2 through 13 apply data cleansing to  $S\_path$  to update  $S\_path$  and generate  $S\_env$  simultaneously. These steps are explained in more detail as follows. In Step 2 and Step 3, the algorithm stores the position data of AIS into a path record of a ship until the ship arrives at certain port or the record of AIS data is not available. In Step 6, the algorithm uses Geohex [17] to index the environment into grids. A Geohex is a grid of hexagons with identical geocodes. In particular, we used a level 11 Geohex (nearly 14 meters between the centers of the grids) in the evaluations described later. In Steps 7 through 11, the algorithm fixes the defective data. In particular, in Step 7, the algorithm amends the destination information of the AIS data

according to the name of the port the ship actually arrived at. In Step 8, the algorithm compensates for lost AIS data. In Steps 9 and 10, the algorithm recalculates the velocity and direction of the ship based on the actual position transition in the record. In Step 11, the algorithm appends information on the future position of the ship. In Steps 7 through 11, the algorithm fixes the defective data. In particular, in Step 7, the algorithm amends the destination information of the AIS data according to the name of the port the ship actually arrived at. In Step 8, the algorithm compensates for lost AIS data. In Steps 9 and 10, the algorithm recalculates the velocity and direction of the ship based on the actual position transition in the record. In Step 11, the algorithm appends information on the future position of the ship.

B. Parameters of CMF

For learning behaviors effectively, the actions that the ship can take should be limited into several discrete options.

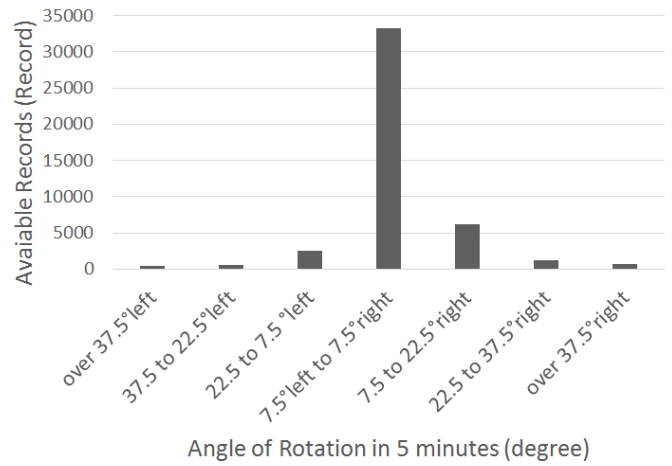


Fig. 5. Distribution of actions in real AIS data.

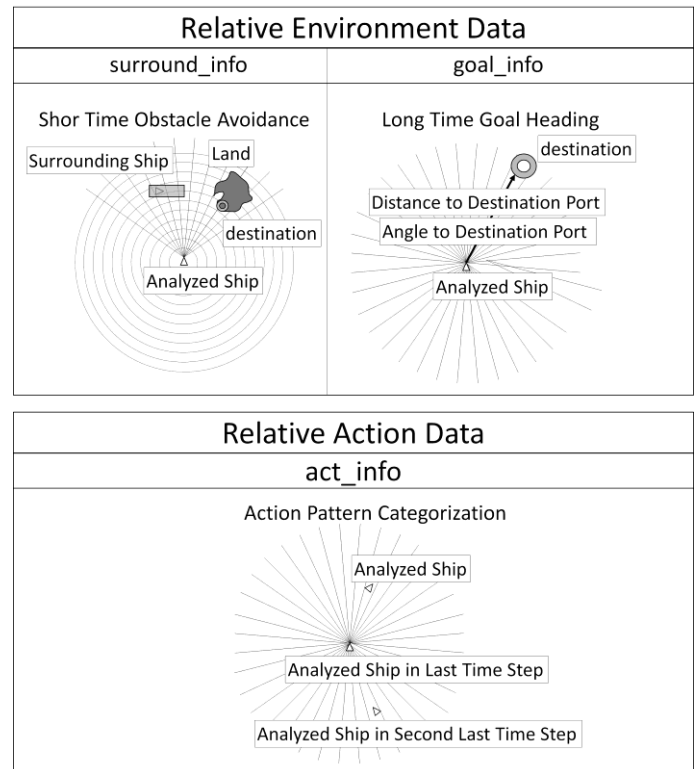


Fig. 6. Definitions of parameters in CMF.

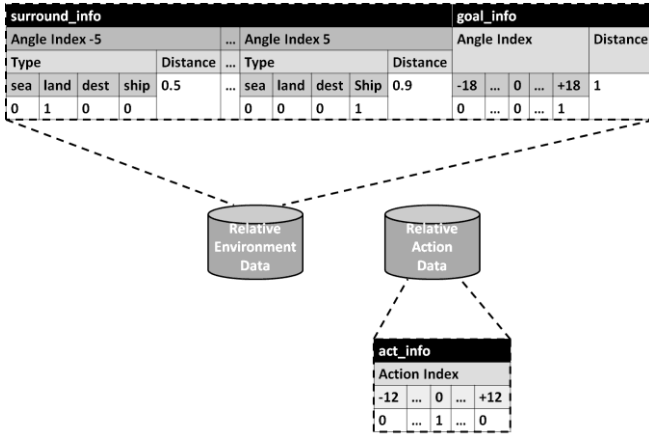


Fig. 7. Data structure of CMF in ship behavior analysis.

Therefore, we set specific parameters for MDP based on the results of the analysis of the AIS dataset.

Fig. 5 shows the distribution of ship behaviors when a ship sails for 5 minutes in Area A, as obtained from our statistical analysis of the cleansed AIS episode data. In 5 minutes, the maximum angle change that a ship can perform is limited to around  $45^\circ$ . The angle disperses when the ship sails for 10 minutes. Therefore, we decided to set the episode length to be 30 minutes and extracted records for 5 minutes from each episode to analyze CMF (we call it a data frame).

Fig. 6 shows how parameters in CMF are defined. We assume that the ship is piloted with considering two factors. One is to avoid obstacles, another is to arrive at the goal.

Therefore, for the data field of the ship's behavior, we defined two main categories of relative environment data: partially observable sections for obstacle avoidance (we call it `surround_info`) and the destination port (we call it `goal_info`). The relative action is calculated from the change in heading of the analyzed ship in 5 minutes (we call it `act_info`).

For convenience of analyzing in angle space, we define an angle index for categorizing the relative angles in the relative environment data, and an action index for relative action data. We define  $\theta$  as the relative angle with respect to the heading of the analyzed ship in the clockwise direction (e.g.,  $\theta = 5^\circ$  means  $5^\circ$  right,  $\theta = -5^\circ$  means  $5^\circ$  left). For the angle index, each angle index subtends an angle of 10 degrees. (e.g., [Index 0:  $-5^\circ < \theta < 5^\circ$ , Index -1:  $-15^\circ < \theta < -5^\circ$ , index 1:  $5^\circ < \theta < 15^\circ$ , etc...]) For the action index, each action index subtends an angle of 15 degrees. (e.g., [Index 0:  $-7.5^\circ < \theta < 7.5^\circ$ , Index -1:  $-22.5^\circ < \theta < -7.5^\circ$ , etc...]).

Fig. 7 shows the data structure of CMF. The data structure of `surround_info`, `goal_info` and `act_info` is explained below.

#### ● Surround\_info

For analyzing surround info, the observation of the surrounding environment is set to the nearest feature elements in the observable range. Observable angle range is defined as  $-45^\circ < \theta < 45^\circ$ . (The index varies from -5 to 5) and is separated into observable sections by the angle index.

The observable distance range is limited to 10 [km] from the analyzed ship. We assign [sea], [land], [ship] and [destination port] as types of features. If [land], [ship] and [destination port] are not in the observable distance range of an observable section, the feature type of the observable section is set as [sea]. The quantified feature is expressed as a one hot vector. The distance to the features is quantified using

a distance conversion function, as follows.

$$Feature\_Dist = \frac{\mathit{math.floor}((1 - \frac{Dist}{10000}) * 100)}{100} \quad (7)$$

The distance to the features is quantified for each observable section. If the observed feature type is [sea], value is set to 0.

#### ● Goal\_info

The `goal_info` is calculated for range of  $-180 < \theta < 180^\circ$ , separated by the angle index over an observable range of 60 kilometers. `Goal_info` calculates the angle gap and distance between the analyzed ship and its destination. The angle gap is the angle between the direction of the analyzed ship's heading and the destination and it is quantified by a one-hot vector of angle indexes. If the distance to the destination exceeds the maximum range: 60 kilometers, the feature value is set to 0. The distance to the destination is

#### Algorithm 2 Making CMF from cleansed AIS dataset

Input: Environment Datasets `S_env`, Path Datasets `S_path`  
Output: Relative Environment Dataset `S_relative_env`,  
Relative Action Dataset `S_relative_action`

function CMF:

- 1: for Episode Data in `S_path`:
- 2: `dataFrameList` =  
    `extractDataFrame(S_path, dataframeLength)`
- 3: for each `dataFrame` for `dataFrameList`:
- 4: `Position_ship` = `getPosition(dataFrame)`
- 5: `Destination_ship` = `getDestination(dataFrame)`
- 6: `goal_info` = `goal_calc(Position_ship, Destination_ship)`
- 7: for each `environmentElements` in `S_env`:
- 8: `Position_elements, Type_elements` =  
    `getInfo(environmentElements)`
- 9: `Distance` = `calculate_distance(Position_ship, Position_elements)`
- 10: `Surround_dict[Type_elements].append`  
    (`get_Surround_Info(Distance, Type_elements)`)
- 11: `surround_info` = `getMinimumfeatures(Surround_dict)`
- 12: `Feature_value` = [`goal_info`, `surround_info`]
- 13: `S_relative_env.append(Feature_value)`
- 14: `Position_future` = `getFutureCoord(dataFrame)`
- 15: `action_info` = `getActionInfo(Position_ship, Position_future)`
- 16: `S_relative_action.append(action_info)`
- 17: return `S_relative_env, S_relative_action`

Fig. 8 AIS data relation calculation algorithm

$$Feature\_Dist = \frac{\mathit{math.floor}((1 - \frac{Dist}{60000}) * 100)}{100} \quad (8)$$

- Act\_info

The act\_info is calculated for a range of  $-180 < \theta < 180^\circ$ , separated by the action index. The action direction of the analyzed ship at  $t = 0$  minute is calculated from the position of the analyzed ship at  $t = 5$  minutes. Here, we assume that there is a change in action (e.g., turn  $15^\circ$  right, do not turn, etc...) for the analyzed ship at  $t = 0$  and simulate its expected position. By comparing the expected position with the observed one, we categorize the action taken by the analyzed ship at  $t = 0$  on the basis of the action index.

### C. Data Conversion

Fig. 8 lists the algorithm to make CMF data from the output of Algorithm 1. This algorithm takes as input the environment datasets  $S_{env}$  and path datasets  $S_{path}$ . It outputs the relative environment dataset  $S_{relative\_env}$  and relative action dataset  $S_{relative\_action}$ . The CMF data is created from consecutive records, i.e., data frames. In Step 1 and Step 2, the data in the path database is converted into data frames. In Steps 3 through 16, data is extracted from data frames instead of episodes. In Steps 4 through 6, the relative angle and relative distance to the destination port is calculated as goal\_info. In Steps 7 through 9, the relative angle and relative distance to features around the analyzed ship are calculated and stored in a temporal dictionary, called Surround\_dict. The type and distance to the closest features in the partial observation range for the analyzed ship is extracted from Surround\_dict in Step 11. In Step 12 and Step 13, goal\_info and surround\_info are integrated and stored in a list. In Steps 14 through 16, action\_info is calculated from the change in the angle of the ship's heading. In this algorithm, data cleansing is applied to the data frame instead of to the episode data in the  $S_{path}$  itself. By using data frames, the actions of the analyzed ship can be efficiently categorized into a limited number of options and thereby the quantity of data that can be dealt with is enlarged. Also, by simply using the information of the nearest features in observable sections, the number of state patterns in MDP is limited. Therefore, the effectiveness of the data analysis is increased.

## VI. EVALUATION

We applied CMF to a dataset of AIS data in Area A. We forecast the behaviors of ships by forecasting the way of the ship ([turn left, keep straight, turn right]) and exact angle of the way by using the action index ([Action Index -3, Action Index -2, ... Action Index 3]).

We developed a DQN to forecast behaviors of ships. The neural network takes surround\_info as input and goal\_info in CMF as feature values  $\phi(s)$  of the environment and outputs the estimated  $Q(s,a)$  value for each action option regularized in the range of 0 to 1 by using the softmax function. The dimension of the output layer is 3 nodes for the way forecasting problem and 7 nodes for the angle forecasting problem. action\_info in CMF is used as an answer in the training of the angle forecasting problem. For the way forecasting problem, an action index of -3 to -1 is categorized as [turn left], and an action index of 1 to 3 is categorized as [turn right]. There are four hidden layers in the DQN. All of the nodes in the hidden layers are fully connected. The

dimensionalities of the hidden layers are 500, 250, 250, and 50, where the hidden layer with 500 nodes is connected to the input layer. Each activation function in the neural network is a Rectifier Linear Unit (ReLU), except that the first hidden layer uses a sigmoid function.

To examine the effect of CMF, we compared forecasting accuracy in our method with a state transition (ST) model developed from the dataset generated by applying only data cleansing and data management to the AIS dataset taken from Area A. The ST model applied a level 11 GeoHex.

The system using our proposed method was 85% accurate on the way forecasting problem and 81.2% accurate on the angle forecasting problem. The system using the ST model had 52% accuracy. Thus, our method improved accuracy in angle forecasting by 29.2%.

## VII. DISCUSSION

This section discusses why CMF raises the accuracy of forecasting compared with traditional datasets.

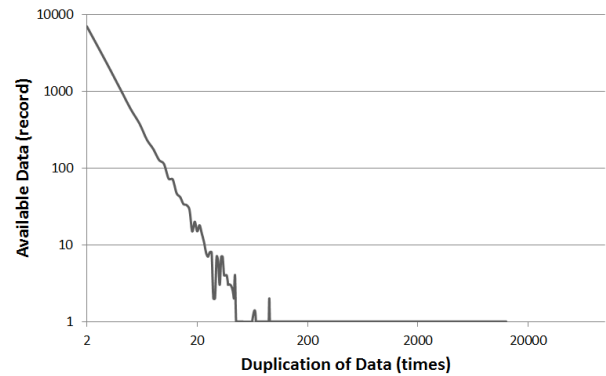


Fig. 9. Data duplication distribution after applying CMF.

TABLE I: SUMMARY OF RESULTS

| Database    | Forecasting Accuracy | Data Usage Rate |
|-------------|----------------------|-----------------|
| Without CMF | 52 [%]               | 27.9 [%]        |
| With CMF    | 81.2 [%]             | 62.1 [%]        |

From the 7307679 records of AIS dataset gathered in 1 month at Area A, we collected 16222 episodes. In this data, 6849 out of 16222 episodes were not observed again within 1 month. In this data, episodes applicable for state transition model are amounted to 4542 episodes. The data usage rate was only 27.9%. These episodes do not have the same situation patterns in them. Thus, the data is not suitable for AL.

By applying CMF, as a result of compressing the available state patterns, relative environment dataset contains similar situations for certain data. We got 150238 data frames from the episodes. 93324 of these data frames contain data in the same situation at least twice. Thus, 62.1% of the data could be used for the analysis. Because multiple actions were observable for the same situation, we can expect that applying CMF will improve the accuracy of AL. Fig. 9 shows the counts of duplicated data for the AIS data set after applying CMF. The X axis of the figure represents the number of times the data is observed in the situation and the y axis shows the patterns collected from the 1-month AIS dataset. For more than 10 situation patterns, we observed over 20 data frames in the same situation. For the most frequent situation, we got

12609 data frames.

To evaluate the accuracy of our system with alternative technology, we compared experiment results of our system with a system using human expert knowledge [18]. According to [18], the experimenter obtained peer-to-peer location information transmitted from ships nearby by Zigbee to forecast their way, and forecasted the ways of the surrounding ships with 80% accuracy. The accuracy of our system (85%) at forecasting the ways of the surrounding ships was superior to that of the human expert using local information from Zigbee. Thus, our method has the potential to support unmanned shipping in crowded bay areas by forecasting the paths of surrounding ships.

### VIII. CONCLUSION

We proposed data management using a data structure conversion named Co-Moving Frame (CMF). Apprenticeship learning (AL) can use CMF to analyze the behaviors of ships as recorded in sparse AIS data because CMF solves data sparseness problem by compression. By applying AL, we successfully forecast paths of ships with 81.2% accuracy on a dataset to which CMF was applied. The forecast was 29.2% more accurate than a state transition model developed from the same dataset without applying CMF. Applying CMF increases data usage effectiveness from 27.9% to 62.1%, by increasing the quality of the data. These results indicate the possibility of creating a support system for unmanned shipping by forecasting the behavior of surrounding ships in crowded sea.

### APPENDIX. APPROXIMATE THE AMOUNT OF GRIDS IN AREA A

The distance between hexagonal grids at level N Geohex at a latitude of  $\theta$  is formulated as

$$\frac{2570813.5190219423}{3^N} \cos(\theta) [m] \quad (9)$$

By assuming  $N = 11$  and  $\theta = 35.4$  North, the distance is approximately 14 meters. This size is quite close to a typical width of the ship. Therefore, we decided to use level 11 Geohex for making grids for Tokyo Bay.

The distance of the sea region in Area A used in the evaluation can be calculated from Heubeny's equation, with the following result.

Width: 43707 meters at North bound

43817 meters at South bound

Length:

22523 meters

The width varies because the shape of the Earth is elliptical. Because the Geohex grid is hexagonal and two sides of the grid are parallel to the Equator, the grid in a certain area can be calculated as follows.

$$\text{Width\_Expectation\_Value} = \frac{\text{Width}}{\text{HexDis} \tan ce} * \frac{2}{3} \quad (10)$$

$$\text{Length\_Expectation\_Value} = \frac{\text{Length}}{\text{HexDis} \tan ce} * \frac{2}{\sqrt{3}} \quad (11)$$

Assuming a Hex Distance of 14 meter, the expectation values of the width and lengths of grids inside Area A are as follows.

$$\text{Width\_Expectation\_Value} = \frac{\text{Width}}{14} * \frac{2}{3} \cong 2083 \quad (12)$$

$$\text{Length\_Expectation\_Value} = \frac{\text{Length}}{14} * \frac{2}{\sqrt{3}} \cong 1858 \quad (13)$$

The expected number of available grids in Area A is thus

$$2083 * 1858 = 3870214 \cong 3.9 * 10^6 \quad (14)$$

### REFERENCES

- [1] H. C. Burmeister, MUNIN The unmanned dry bulk carrier. [Online]. Available: <http://www.unmanned-ship.org/munin/wp-content/uploads/2014/09/MUNIN-WS@SMM-140909-3-Introduction-to-MUNIN-HCB.pdf>
- [1] International Maritime Organization, "Guidelines for the installation of a shipborne automatic identification system(AIS) ," *International Maritime Organization*, January 2003
- [2] U. Simsir, M. F. Amasyali, M. Bal, U. B. Çelebi, and S Ertugrul, "Decision support system for collision avoidance of vessels," *Applied Soft Computing*, vol. 25, pp. 369-378, August 2014
- [3] R. Zhang and M. Furusho, "Constructing a decision-support system for safe ship-navigation using a bayesian network," *Digital Human Modeling: Applications in Health, Safety, Ergonomics and Risk Management DHM 2016*, Lecture Notes in Computer Science, vol. 9745, Springer
- [4] J. George, J. Crassidis, T. Singh, and A. M. Fosbury, "Anomaly detection using context-aided target tracking," *Journal of Advanced Information Fusion 2011*, vol. 6, pp. 39–56, June 2011
- [5] T. Shirai, N. Kubo, K. Inada, and H. Tamaru, "A basic study on prediction of ship status at Tokyo bay using traffic flow network," *The Journal of Japan Institute of Navigation*, vol. 135, 2016, pp. 108-114, December 2016.
- [6] P. Abbeel and Y. A. Ng, "Apprenticeship learning via inverse reinforcement learning," Paper presented at 21st International Conference on Machine Learning, Banff, Canada, 2004
- [7] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep Q-networks," *NIPS workshop on Deep Learning for Action and Interaction 2016*, Barcelona Spain, Neural Information Processing Systems Foundation, Inc., 2016
- [8] R. C. Anthony, "Exact and approximate algorithms for partially observable Markov decision processes," Ph.D. Dissertation. Brown University, 1998, Providence, RI, USA
- [9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop 2013*, Nevada US, Neural Information Processing Systems Foundation, Inc., 2013
- [2] TST Corporation Retrieved November 16. (2017). [Online]. Available: <http://www.toyoshingo.co.jp/site/portal/en/index.html>
- [10] M. Minami and R. Shoji, "Collision hazard evaluation using AIS in Tokyo bay," *The Journal of Japan Institute of Navigation*, vol. 133 pp. 81-87, December 2015.
- [11] Review of Maritime Traffic Regulations for Tokyo Bay and Ise Bay, White Paper on Traffic Safety in Japan 2008, p. 31-33, 2008
- [12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," Paper presented at 12th European conference on Computer Vision - Volume Part IV (ECCV'12), 2012
- [13] Revised Guidelines for Onboard Operational Use of Ship Automatic Identification Systems, 2016
- [14] A. Fluit, "AIS information quality report of static AIS messages: "AIS information quality report" region: HELCOM, farvandsvesenet," Danish Maritime Safety Administration, 2011
- [15] Geohex. Retrieved November 16. (2017). [Online]. Available: <http://geohex.net/>
- [16] T. Yamaguchi and S. Kubota, "A collision prevention system for small crafts employing ZigBee sensor network," paper presented at IEICE

Information and Communication Engineers Student Association in Tokyo, Research Presentation, 2016.



**Yuxin Liang** received a bachelors degree in electrical engineering from the University of Washington, WA in 2013 and a masters degree in information science and technology from the University of Tokyo, Japan in 2015.

He has been working for the Research and Development Group, Hitachi, Ltd. as a Researcher since 2015.

Mr. Liang is a member of the Institute of Electronics, Communication Engineering (IEICE).



**Masayoshi Mase** received a Ph.D. in engineering from the Department of Computer Science and Engineering, Waseda University, Tokyo, Japan in 2011.

He worked for the Department of Computer Science and Engineering, Waseda University as a Research Associate in 2008-2011 and has been working for the Research and Development Group, Hitachi, Ltd. as a Researcher since 2011.

Dr. Mase is a member of the IEEE Computer Society, ACM, and Information Processing Society of Japan (IPSJ).