

Probabilistic Modeling of Mobile Agents' Trajectories

Štěpán Urban, Michal Jakob and Michal Pěchouček

Agent Technology Center, Dept. of Cybernetics, FEE Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{urban, jakob, pechoucek}@agents.felk.cvut.cz

Abstract. We present a method for learning characteristic motion patterns of mobile agents. The method works on two levels. On the first level, it uses the expectation-maximization algorithm to build a Gaussian mixture model of the spatial density of agents' movement. On the second level, agents' trajectories as expressed as sequences of the components of the mixture model; the sequences are subsequently used to train hidden Markov models. The trained hidden Markov models are then employed to determine agent type, predict further agent movement or detect anomalous agents. The method has been evaluated in the maritime domain using ship trajectory data generated by the AGENTC maritime traffic simulation.

Keywords: trajectory modeling, spatio-temporal learning, mobile agents, maritime traffic

1 Introduction

Trajectories represent an important external manifestation of mobile agents' behavior. Modeling agents' trajectories therefore provides a valuable tool applicable in a number of scenarios ranging from categorizing unknown agents, predicting their future movement (and possibly action) up to identifying agents acting in an unusual and potentially malicious way.

In this paper, we propose a method which captures the spatial and temporal aspects of agents' movement using two different representations, a Gaussian mixture model and a hidden Markov model. This decomposition allows to reduce the computational complexity of the overall learning problem.

Our work is inspired by the work on motion pattern learning (e.g. [10,7]) as well as work on agent behavior modeling (e.g. [4,14]). The proposed method is general, though it has been developed as part of our larger work on modeling, detecting and disrupting illegal maritime activities [9,8].

After introducing our two-level model in Section 2, we show how it can be employed in two specific classification scenarios in Section 3. Evaluation on the domain of maritime traffic is presented in Section 4. Section 5 summarizes related work and we conclude with final remarks in Section 6.

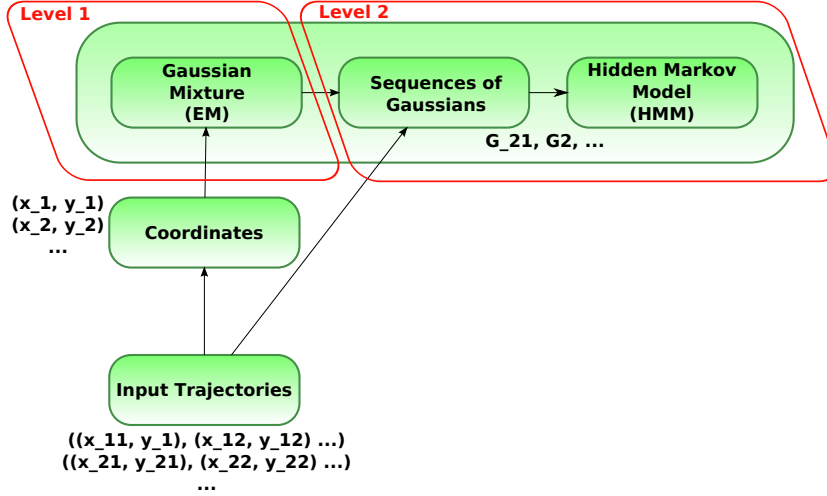


Fig. 1. Schema of the two-level trajectory modeling method

2 Two-level Trajectory Model

As already mentioned, our approach represents agents' movement at two levels.

The proposed method works on two levels. On the first level, spatial properties of the traffic are represented using a Gaussian mixture model (GMM). On the second level, temporal aspects are captured using a hidden Markov model (HMM).

The input to the algorithm is a set of agent trajectories $\mathcal{T} = \{T_1, \dots, T_m\}$. Each trajectory $T = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ is a tuple representing a sequence of 2d coordinates $\mathbf{x} = (x_{lat}, x_{long}) \in \mathbb{R}^2$. The output of the algorithm is a Gaussian mixture model (Level 1) and a hidden Markov model (Level 2) best approximating the trajectories. Figure 1 gives a graphical overview of our approach.

2.1 Spatial Modeling

On the first level, the method uses the expectation-maximization algorithm [6] to build a Gaussian mixture model of the spatial density of the agents' motion. The algorithm disregards the sequential aspect of the agents' trajectories and treats them as unordered sets of agent positions. It then approximates the empirical distribution of these positions using a mixture of 2d Gaussian kernels. The process can be viewed as clustering the very large number of agents' positions into a significantly lower number of spatial clusters which are then used as the basis for temporal modeling.

More specifically, each Gaussian kernel

$$\phi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (1)$$

is parameterized by its mean $\boldsymbol{\mu} \in \mathbb{R}^2$ (the position of Gaussian kernel's center) and its 2-by-2 covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2,2}$. Assuming the mixture consists of m components, the algorithm is looking for a tuple of parameters

$$\Theta = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, w_1, \dots, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, w_m)$$

so that the mixture model

$$\Phi(\mathbf{x}|\Theta) = \sum_{i=1}^m w_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2)$$

best approximates the spatial distribution of the agents' positions in the trajectory set \mathcal{T} . The expectation-maximization algorithm [6] is used to find maximum likelihood estimates of parameters Θ^* , i.e.,

$$\Theta^* = \arg \max_{\Theta} \Phi(\mathcal{T}|\Theta) = \arg \max_{\Theta} \prod_{T=(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{T}} \prod_{i=1}^n \Phi(x_i|\Theta) \quad (3)$$

for a given set of trajectories \mathcal{T} . An example of such a set is given in Figure 2.

The set of Gaussian kernels obtained is used as a basis for expressing agents' trajectories in the second level of the algorithm.

2.2 Temporal Modeling

The second level captures the temporal structure of agents' trajectories. Agents' trajectories are expressed as sequences of the Gaussian components of the mixture model; the sequences are subsequently used to train hidden Markov model.

Specifically, assume we have a Gaussian component model Φ (see (2)) and a trajectory $T = (\mathbf{x}_1, \dots, \mathbf{x}_n)$. For each point \mathbf{x} , the algorithm looks for a Gaussian kernel component to which the point belongs

$$i^* = \arg \max_{1 \leq i \leq m} \phi(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (4)$$

By applying the above to all points, we obtain a sequence $Q = (i_1, \dots, i_n)$ expressing the original sequence of locations as a sequence of (the indices of) the components of the GMM; we term such a sequence a *GMM-based trajectory*. After applying the above to all trajectories \mathcal{T} , we obtain a set of GMM-based trajectories \mathcal{Q} .

In the next step, a hidden Markov model is sought which best fits the sequences in \mathcal{Q} . We assume that the states of the model are observable and directly correspond to the components of the GMM-based model. The *Baum-Welch algorithm*[2] is then applied to learn a set \mathcal{P} of transition probabilities $P(i_j|i_k)$ specifying the probability that an agent moves from a geographical region corresponding to component j to a region corresponding to component k .

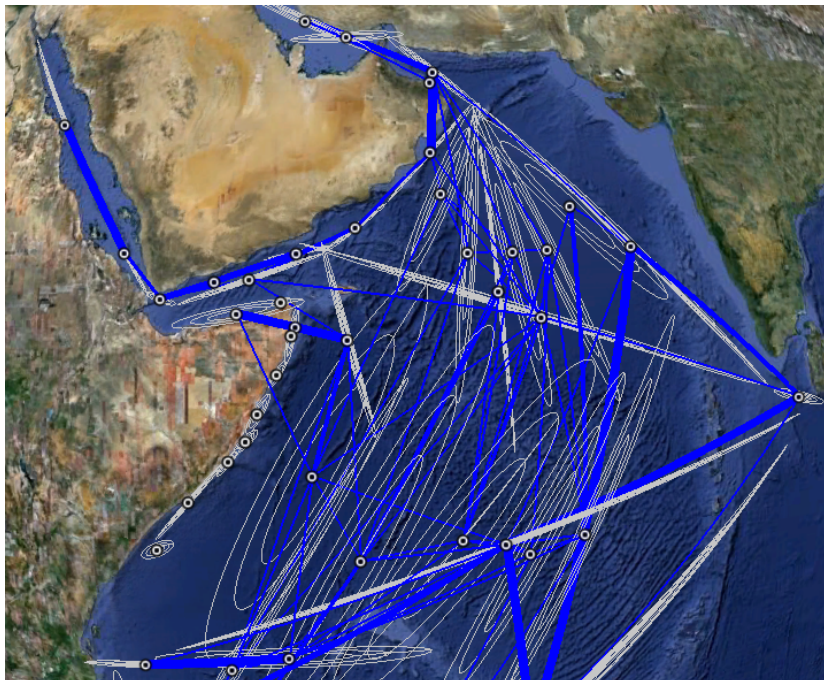


Fig. 2. Example two-level model. The white ellipsoids represent Gaussian kernels in the Level-1 GMM; blue overlay graph represents the Level-2 hidden Markov model. The thickness of the edges corresponds to the transition probability between the two kernels, i.e., between regions they represent.

Once a hidden Markov model is obtained, it can be used for evaluating the closeness of specific trajectories. Also in this case, the classified trajectory $T = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ is first converted into its GMM-based representation $Q = (i_1, \dots, i_n)$ using (4). The probability of the sequence Q being produced by a model \mathcal{P} is then calculated as

$$p(Q|\mathcal{P}) = \prod_{j=2}^n P(i_j|i_{j-1}) \quad (5)$$

An example of a HMM obtained using the outlined method is given in Figure 2.

3 Classification Modes

The two-level model can be employed in two modes (1) for identifying anomalous and thus possibly illegitimate agents, and (2) for categorizing agents into a predefined set of classes.

3.1 Agent Categorization

The agent categorization mode assumes there are labeled trajectory sets available for all categories of agents under consideration. On Level 1, all trajectories are used for creating a single spatial model of the agent traffic; this is shared across all categories. On Level 2, an individual HMM for each category is created.

When an unknown agent is to be classified, it is evaluated for closeness against all HMMs (using (5)) and the category of the closest HMM is used as the category of the agent.

We refer to the classifier with the above described structure and operation as the *agent categorizer*.

3.2 Illegitimate Agent Detection

The illegal agent detection mode assumes that only trajectories for legitimate agents are known (and labeled); there are no known trajectories for illegitimate agents¹. The learning phase of the algorithm is similar to the agent categorizer, only now HMMs are created only for legitimate agents.

When an unknown agent is to be classified, it is evaluated for closeness against all HMMs. However, because the HMMs now do not cover all categories of agents, the agent is classified as "legitimate" only if the closeness of its trajectory to the closest HMM is higher than a defined *closeness threshold*. Otherwise, the agent is categorized as "illegitimate". By varying the closeness threshold, we can moderate the trade-off between false negatives (an anomalous agents is classified into one of the legitimate classes) and false positives (a legitimate vessel is classified as anomalous); the trade-off can be quantified using an ROC curve.

We refer to the classifier with the above described structure and operation as the *illegitimate agent detector*.

4 Experimental Evaluation

We have evaluated the proposed method in the domain of maritime traffic. Vessel trajectory data were obtained from a data-driven simulation utilizing real-world data sources and FSM-based agent behavior models.

4.1 Experimental Testbed

The testbed used for evaluation has been developed as part of our larger activity on applying agent-based techniques to fight maritime crime. At the center of the testbed is an agent-based simulation platform incorporating a range of real-world data sources in order to provide a solid computational model of maritime activity. The platform can simulate the operation of thousands vessels of the following type:

¹ This setting is often referred to as *learning from positive examples only*

- **Long-range transport vessel** – large- to very large-size vessels transporting cargo over long distances (typically intercontinental); these are the vessels that are most often targeted by pirates.
- **Short-range transport vessel** – small-to medium-size vessels carrying passengers and/or cargo close to the shore or across the Gulf of Aden.
- **Fishing vessel** – small- to medium-size vessels performing fishing within designated fishing zones; fishing vessels launch from their home harbors and return back after the fishing is completed.
- **Pirate vessel** – medium-size vessels operating within designated *piracy zones* and seeking to attack a long-range transport vessel. The pirate control module supports several strategies some of which can employ multiple vessels.

The behavioral models for individuals categories of vessels have been synthesized from the information about real strategies [1]. The vessel operational characteristics (length, tonnage, max speed etc.) are based on real-world data² too. More information about the testbed including a brief video overview can be found in on project’s web site³ and in [8].

4.2 Experimental Data

The simulation was run for 3215 simulation seconds and the trajectories sampled with 10-minute resolution. All four vessel classes implemented by three different FSMs were active in the simulation. The experimental scenario contained 500 vessels of each class. The execution of the simulation produced traces containing altogether 3,588,909 coordinates which were used for all subsequent analyses. A subset of produced trajectories is shown in Figure 3.

4.3 Method Configuration

A key parameter of the method is the number of Gaussian kernels used by the Level 1 mixture model. To determine how many Gaussian distributions should be used to approximate agent spatial distribution with sufficient accuracy, we run the EM algorithm for different number of components C (from $C = 1$ to $C = 100$). Low number of Gaussian distributions cannot approximate real distribution well enough; on the other hand, for high number it will be difficult to learn the HMM (on level 2) because HMM learning time is increasing exponentially with number of hidden states. To see how well a GMM fits the data, we observe the likelihood function used in EM algorithm. From the likelihood function plot shown in Figure 4, we determine that we need at least 40 Gaussian distributions, which is the value used in our experiments.

² e.g. <http://aislive.com>, <http://vesseltracker.com>

³ <http://agents.felk.cvut.cz/projects/agentc/>



Fig. 3. An subset of vessel trajectories used in the evaluation of the method (while lines)

Probabilities of all plausible transitions in the HMMs were initialized with small non-zero positive value (0.01) to avoid the case where sequences (especially long ones) are classified as having a zero degree of membership to a given HMM just because a particular transition was not present in the training data. This initialization procedure results in a significantly higher accuracy of classification.

4.4 Results

We now present the results for both classification modes.

4.5 Results for Agent Categorization

Agent categorization mode described in 3.1 was used to classify agents on experimental data described in 4.2. The classification accuracy was evaluated using 10-fold cross-validation. The resulting dependency of the accuracy on the length of test sequences, i.e. the sequences being classified, is given in Figure 5. In order to gain better insight into the operation of the classifier, we have also calculated the confusion matrix (Table 1).

We also calculate the confusion matrix for the classifier to understand the structure of misclassifications. The confusion matrix is given in Table 1. It follows

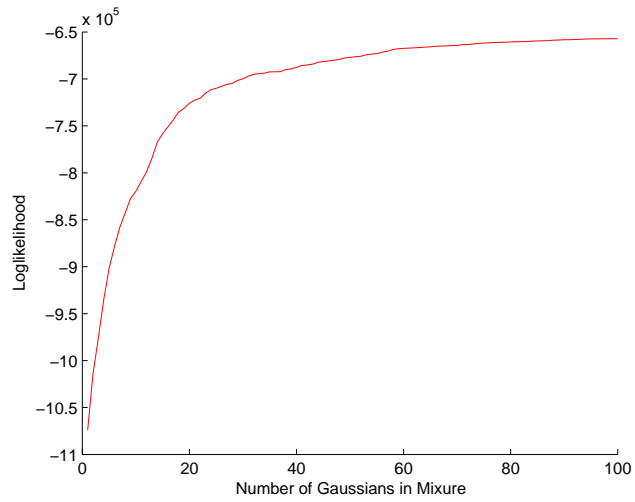


Fig. 4. Log-likelihood function in the dependence of number of the Gaussian kernels in the Level 1 model

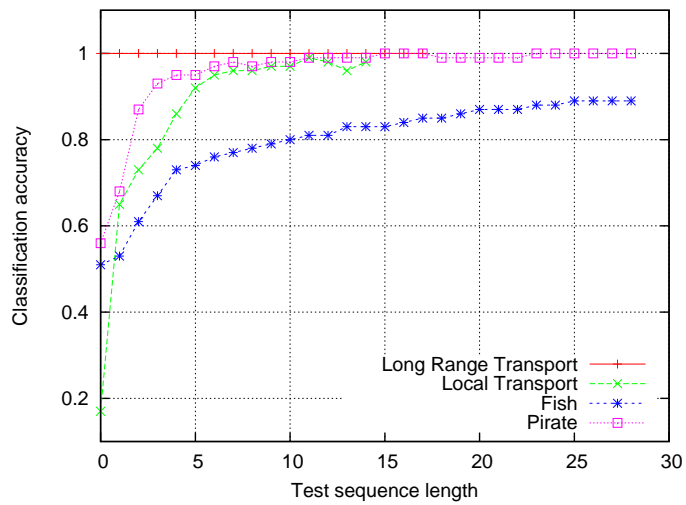


Fig. 5. Classification accuracy of the agent categorization classifier for different trace lengths

that long-range transport vessels are the most easily identifiable; this is because they have completely different trajectories and visit different locations than other vessels. Local transport vessels are sometimes misclassified with fishing vessels (and vice-versa); that is because both classes of vessels operate close to the coast and their trajectories overlap. For similar reasons but with lower rate, pirate vessels are sometimes misclassified as local transport.

It follows that long-range transport vessels are the most easily identifiable; this is because they have completely different trajectories and visit different locations than other vessels. Local transport vessels are sometimes misclassified with fishing vessels (and vice-versa); this is because both classes of vessels operate close to the coast and their trajectories overlap. For similar reasons but with lower rate, pirate vessels are sometimes misclassified as local transport.

Predicted	Actual			
	Long-range transport (1)	Local transport (2)	Fishing (3)	Pirate (4)
(1)	1	0	0	0
(2)	0	0.72	0.18	0.01
(3)	0	0.20	0.77	0.03
(4)	0	0.08	0.05	0.95

Table 1. Confusion matrix of the agent categorizer

4.6 Results for Illegitimate Agent Detection

For the detection classifier (Section 3.2), only data corresponding to the three legitimate types of vessels (long-range transport, short-range transport and fishing) were used. Again, 50 Gaussian kernels were used in the Level 1 model.

The accuracy of classification into legitimate vs. illegitimate agent was again evaluated using 10-fold cross-validation. As noted above, the trade-off between false negatives and false positives can be regulated by setting the closeness threshold of the classifier. By modifying the similarity threshold, we can moderate the trade-off between the false positive (an anomalous agent is classified into one of the legitimate classes) and false negative (a legitimate agent is classified as anomalous) rate. The trade-off can be quantified using an ROC curve⁴ in Figure 6 (blue crosses). A test with perfect discrimination (no overlap in the two distributions) has a ROC plot that passes through the upper left corner (100% sensitivity, 100% specificity).

⁴ *Receiver Operating Characteristic* (ROC), or simply *ROC curve*, is a graphical plot of the sensitivity vs. (1 - specificity) for a binary classifier system as its discrimination threshold is varied.

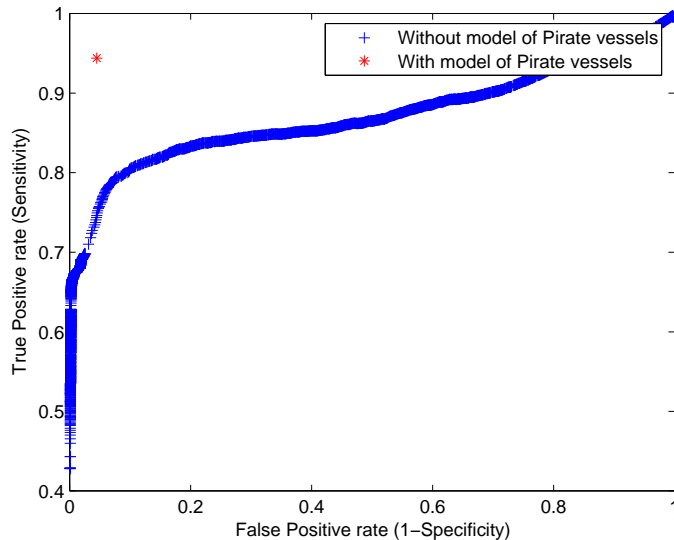


Fig. 6. ROC curve for the legitimate agent classifier (for a varied similarity threshold under which 'unknown/anomalous' classification output is produced). In addition, sensitivity vs. specificity relation for the agent categorizer classifier is also depicted (single point only as there is no variable parameter affecting the relation).

From the confusion matrix, we can compute sensitivity and specificity and compare this measure with the values obtained by the legitimate agent detection classifier. This value is only one number in ROC curve 6 and is depicted by a red cross. The position of the cross in the upper left corner shows the superior performance of the all class classifier over the illegitimate agent classifier described previously. This is not surprising given that the all class classifier model is constructed from a more representative learning base.

5 Related Work

Perhaps the approach most similar to our work is described in [10] where agent trajectories are represented as sequences of flow vectors, each vector consisting of four elements representing position and velocity of the object in 2D space. The patterns of trajectories are formed by a neural network. In [11], the authors use the EM algorithm for learning zones. To learn agent traces, they simply compare a new trajectory with all routes that are already stored in the database using a simple distance measure. The limitation of this method is that they only use spatial information and temporal information is not well represented. In [7], the

authors use a system based on a 2-layer hierarchical version of fuzzy K-means clustering. They first cluster similar agent trajectories into the same cluster according to their spatial information. Each object in a spatial cluster is then clustered according to temporal information. In addition, there are methods for detecting anomalies by direct comparison of behaviors without learning agent's motion patterns [5].

The specific application of agent motion patterns modeling on vessel agents has been studied in past few years, with existing papers mostly focused on anomaly detection. In [13], the authors use the framework of adaptive kernel estimation and hidden Markov models for the purpose of anomaly detection. In [3] and [12], the authors use a neural network trained on space-discretized AIS trajectory data to learn behavior of normal agents and then to detect anomalies as well as to predict future vessel position and velocity.

6 Conclusion

We have proposed a method for learning motion patterns of mobile agents. The combination of spatial modeling using GMM (level 1) and temporal modeling using HMM (level 2) provides expressive framework for capturing agent behavior. We provide variants both for both the case where learning data are available only for the legitimate agents and for the case where data are available for all categories of agents. In the former case, the mechanism works as an anomaly detector – agent traces which are not similar enough to any of the legitimate models are classified as anomalous and thus likely illegitimate. In the latter case, the mechanism works as a standard classifier. We have evaluated the method in the maritime domain using trajectory data of four different categories of ships.

The future work will proceed along several possible directions. On the technical side, more research into the operation of the individual modeling algorithms at both levels and their mutual interplay is needed, including experimentation with alternative frameworks for spatio-temporal pattern representation. On the modeling side, the introduction of more background and context information should help to further improve the accuracy of the produced models.

7 Acknowledgements

The work presented is supported by the Office for Naval Research project no. N00014-09-1-0537 and by the Czech Ministry of Education, Youth and Sports under Research Programme no. MSM6840770038: Decision Making and Control for Manufacturing III.

References

1. IMB Piracy Reporting Centre. http://www.icc-ccs.org/index.php?option=com_content&view=article&id=30, 2009.
2. L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.
3. N. Bomberger, B. Rhodes, M. Seibert, and A. Waxman. Associative learning of vessel motion patterns for maritime situation awareness. In *Information Fusion, 2006 9th International Conference on*, pages 1–8, July 2006.
4. D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 62–67, Portland, Oregon, 1996.
5. H. Z. Computer and H. Zhong. Detecting unusual activity in video, 2004.
6. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
7. W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *Pattern Analysis and Machine Intelligence, IEEE transactions on pattern analysis and machine intelligence.*, 28(9):1450–1464, Sept. 2006.
8. M. Jakob, O. Vaněk, Š. Urban, P. Benda, and M. Pěchouček. Employing agents to improve the security of international maritime transport. In *Proceedings of the 6th workshop on Agents in Traffic and Transportation (ATT2010)*, May 2010.
9. M. Jakob, O. Vaněk, Š. Urban, P. Benda, and M. Pěchouček. AgentC: Agent-based testbed for adversarial modeling and reasoning in the maritime domain. *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, May, 2010.
10. N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC '95: Proceedings of the 6th British conference on Machine vision (Vol. 2)*, pages 583–592, Surrey, UK, UK, 1995. BMVA Press.
11. D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(3):397–408, June 2005.
12. B. Rhodes, N. Bomberger, and M. Zandipour. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, July 2007.
13. B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In *Information Fusion, 2008 11th International Conference on*, pages 1–7, 30 2008–July 3 2008.
14. G. Sukthankar and K. Sycara. Policy recognition for multi-player tactical scenarios. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.