Overhead Projection Approach For Multi-Camera Vessel Activity Recognition

George E. Strauch
Data Lab
Texas State University
San Marcos, TX, U.S.A.
ges71@txstate.edu

Jiajian (Jax) Lin
REU SCC
University Of California
Los Angeles, CA, U.S.A.
jaxlin@g.ucla.edu

Jelena Tešić
Computer Science
Texas State University
San Marcos, TX, U.S.A.
jtesic@txstate.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671274

***Abstract*—On-board video sensors on large ships capture video data at high rates and provide real-time object tracking for maritime applications such as piracy and illegal fishing. High volumes of collected video and imagery data require advanced technology to analyze the video feeds, reduce data smog, and alert the crew on ships or coastal guard when unusual activities are detected. We present an integrated end-to-end system that *analyzes* multi-camera ship video feed; *localizes* maritime vessels in the video feed; *identifies* the maritime vessel over multiple cameras; *maps* the vessel track onto an overhead plane; and *identifies* anomalous vessel movement around the ship. In this paper, we focus on a specific activity detection approach in maritime vessel overhead tracks and on synthetic data generation to realistically model maritime boat movements around onboard ship cameras using real-world examples. We propose and compare three novel modes of trajectory analysis and activity classification, using Computing with Words (CWW), a Markov trajectory feature classifier (MTFC), and Naïve Bayes Radial Classifier (NBRC) to detect the activity of vessel approaching the ship, vessel chasing another vessel, and vessel circling around the ship.**

***Index Terms*—maritime, activity, large video feeds, multiple cameras, classification**

I. Introduction

Overhead imagery analytics are gaining importance in integrated maritime surveillance due to an increase in naval traffic, a decrease in crews on the decks of large ships, and an increase of maritime piracy and illegal fishing. The relevant authorities survey the assigned territories to ensure large ship safety, prevent piracy, prevent ships from carrying illegal firearms and smuggling, and prevent illegal, unregulated, and unreported fishing. Piracy is one of the most prominent threats faced by the global shipping industry today. It is estimated that maritime piracy attacks have cost the industry billions of dollars, and these attacks pose significant danger to the crew members aboard. As technology progresses, the use of visual data feeds provides an increasingly viable method of identifying potential threats along with the other onboard sensors.

In general, maritime domain surveillance is carried out using a combination of Automatic Identification Systems (AIS), Coastal Radar Systems (CRS), and Long Range Cameras (LRC) [1]. These systems produce petabytes of video feeds, which cannot be feasibly analyzed in real time. The recent rise of maritime piracy and attacks on transportation ships has cost the global economy several billion dollars. To counter the threat, researchers have proposed agent-driven modeling to capture the dynamics of the maritime transportation system and to score the potential of a range of piracy countermeasures. In the past couple of years, several analytic systems have been proposed to provide and analyze ship path planning, identify suspicious ships, track ships, and approach and attack target ships using video sensor feeds The work has shown promising results as it focuses on heterogeneous information source fusion and uses out-of-box machine learning approaches designed for commercial rather than maritime applications [1], [2]. These systems can greatly benefit from real-time warnings. Visual data feeds from on-board ship cameras are used by the crew only after they have been alerted of a possible attack. Modern surveillance systems can greatly benefit from automated warnings (e.g., ”boat is approaching”, ”boat is circling the ship”) before the crew even spots the boat.

A. Related Work

Autonomous ships are expected to improve the level of safety and efficiency in future maritime navigation, as the autonomous situational awareness task is moved to ships. State-of-the-art surveys consider the data integration of Global Navigation Satellite System (GNSS) receivers and inertial measurement units (IMU), visual and audio sensor data feeds, remote-sensing (RADAR and LiDAR) data, and auxiliary data (Automatic Identification System (AIS) and external data archives) [3]. Octavian et al. proposed an intelligent system that provides and analyzes ship path planning, identifies and track ships. Artificial intelligence-based surveillance systems are integrated in a single Intelligence Coastal Surveillance system for monitoring and analysis, tracking, and enforcement [1]. Bloisi et al [2] proposed a modular system for intelligent maritime surveillance, with a focus on fusing the information from heterogeneous sources. The Vessel Traffic Service System mainly relies on video feeds, and vessel tracking is enhanced when camera video data is used in combination with radar data and the Automatic Identification System (AIS) [2]. L. Patino et al. demonstrated that the onboard sensor information, combined with intelligence from external sources, proved valuable for early piracy threat detection [4]. The team developed a detailed playbook for the anomalous activities of maritime vessels surrounding the ship that we use as

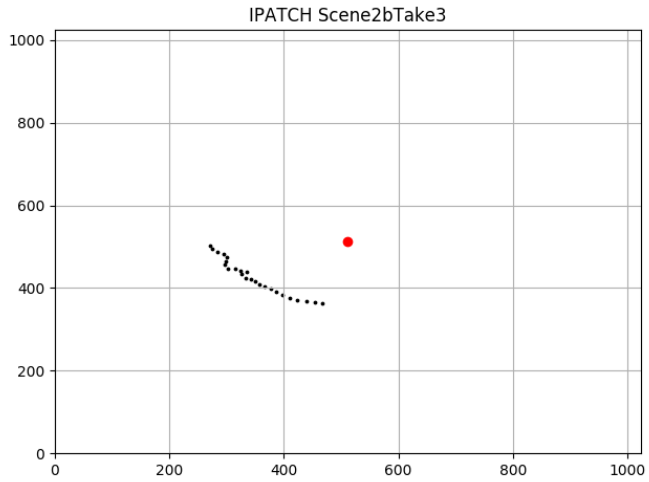

Fig. 1: Sample real-world trajectory detected from multiple camera feeds in IPATCH data [4].

a basis in this paper [5]. The 2020 survey [3] overviews suitable sensors and relevant AI techniques for an operational sensor system. The perception tasks are related to well-defined problems, such as situational abnormality detection, vessel classification, and localization; these problems are solvable using AI techniques, and machine learning methods such as deep learning and Gaussian processes are identified as especially relevant for these problems without a deep dive into actual data smog reduction performances [3]. The 2021 Survey of maritime vessel re-identification, tracking, and multi-model data fusion with data from visual sensors provides the first comprehensive review of research into the use of deep learning in situational awareness of the ocean surface, and it provides a better overview of state-of-the-art methods in maritime vessel research [6].

B. Paper Overview

In this paper, we outline the considerations on design of an end-to-end system to generate automated warnings, and focus on the best machine learning approach adapted to maritime activity and trained to capture boat activities. Section II outlines the video to maritime vessel track procedure, Section III describes three activity recognition approaches, Section IV describes the synthetic data generation; Section V presents the experiments and the measurable results, and Section VI concludes the work and outlines the next steps.

II. End-To-End Activity Identification Pipeline from Multiple Ship Cameras

Camera sensors are inexpensive but the automatic extraction of information from multiple video data streams requires expensive human engagement. First, we design and train algorithms to localize, identify, and track small maritime objects under varying conditions (e.g., a snowstorm, high glare, night) [7], [8]. Next, we produce a model that can use multiple video feeds to map detected objects from an overhead view and classify its path type in a way that can be used to preemptively identify anomalous behavior [9]. We merge the overhead projection from multiple cameras as we know the position of the cameras e.g. as specified in IPATCH [5] data in Figure 2. Each individual scene is composed of at most four distinct cameras: camera twelve faces the stem of the boat with cameras ten, eleven, and fourteen facing starboard and slightly towards the bow. We merge the identified maritime vessels by matching the overlapping detection region of the cameras as in Figure 2, and produce a unique multi-camera track in Figure 1.

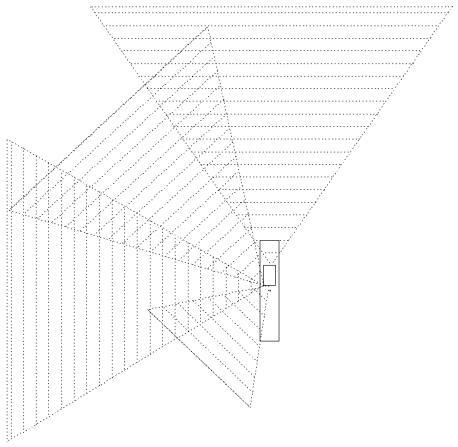

Fig. 2: Connecting tracks over multiple cameras based on location

When analyzing real-world data, we need to consider that any activity we want to detect will probably not occur immediately when the boat is first observed. Moreover, the activity could take place over a large variety of time domains. For example, a potential threat may circle around the ship over the course of hours, subtly change position, and suddenly approach to commence an attack in a matter of minutes or seconds. Therefore, it will be necessary to normalize the path length to make an accurate classification. This can be done by iterating backwards over the previous points and calculating the distance between them until we reach a path length threshold.We would further smooth the path this path section into a constant number of points using means clustering which would make it less noisy and easier for a model to use. Figure 1 shows the result of this done in practice using videos from the IPATCH dataset [4]. The expected boat paths are circling, chasing another boat, approaching, and random, as derived from the real dataset [5]. Figure 2 is an example of an overhead model of a ship that shows the camera positions from Scene 2, Take 3 in the IPATCH data set [4], and an example of how to map the boat's multi-camera defections to the overhead view [9] by using perspective transformation to map the object's center pixel (x,y) coordinate to the overhead model. If the detected object is seen in multiple overlapping video feeds, we may see a slight difference in the calculated location, in which case we define its position as the average (x,y) coordinate on

the overhead model. The result is a set of tracks corresponding to a maritime vessel in the vicinity of the ship.

III. Activity Modeling Approaches

Our work builds on previous work that used Markov trajectory feature classifier (MTFC) [10] and the Computing With Words approach [11], and proposes novel Naïve Bayes Radial Classifier (NBRC) for maritime vessel track activity recognition.

A. Computing with Words

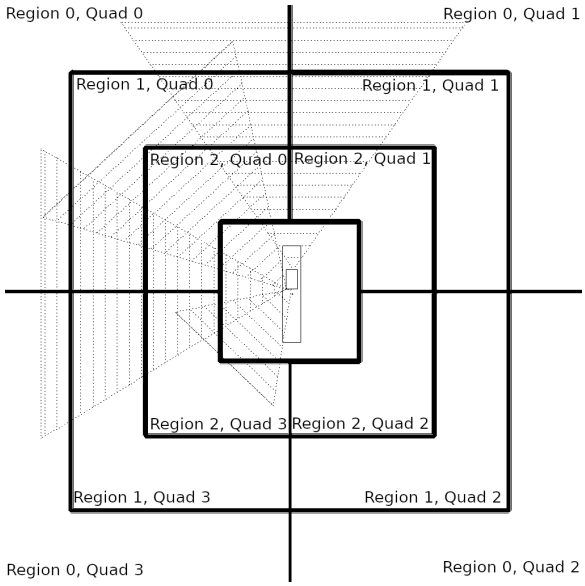

Fig. 3: Computing with Words Regions w.r.t. ship in the center.

Computing With Words (CWW) approach is fast and does not require training data, and its accuracy varies largely based on the trajectory parameters [11]. We implemented Computing with Words (CWW) model [11] as a baseline. This method assigns alphabetical characters to regions that are enclosed around the asset ship in an overhead model. Each of these regions is broken into 4 quadrants that are separated halfway along each axis and represented by a number $0-3$, as illustrated in Figure 3. Computing with Words modeling steps are as follows: (1) within each region, each quadrant is represented by 2 tuples that contain the top left corner pixel location, the bottom right corner pixel location, the quadrant number, and the region character of a rectangular box within the overhead domain. The 2 boxes have an overlapping corner and form an 'L' shape that is one quadrant of a region. The overlap in the quadrant is not an issue since they both represent the same quadrant and region. A total of 8 boxes form a region, and we can vary the thickness and the number of the regions, as illustrated in Figure 3. Each time a region is generated, the tuple objects that define the quadrants are added to a list. When the next region is generated, padding based on the region thickness is added to the corners so that the next region generated is closer the asset ship. To determine in which quadrant a given point belongs, we iterate through the list of tuple objects and find the one where the x value of the pixel location of the point is between the x values for the 2 corners and do the same with the y values. When evaluating a trajectory, we find the location of each point in the model and build a decodable string representing how the trajectory passes through the regions and quadrants. On each new point, we first append the region character only if it changes, and then the quadrant number if it or the region changes. When iterating over the string for a given path, we know the current region was the last alphabetical character encountered, and the current quadrant is the last numerical character encountered. Each activity type meets specific conditions. If the path passes through most of the regions while remaining in one quadrant, it indicates chasing if that quadrant is 1 or 2; if the quadrant is 3 or 4, it indicates approaching. If the path passes through more quadrants than regions, it is circling. If the path passes through at least 2 regions and quadrants, then it is a random path. It is possible for none of these conditions to be met, in which case the track remains as default where no classification is made.

B. Markov Trajectory Feature Classification

Here, we expand on the idea presented in [10], and propose a Markov trajectory feature classifier (MTFC). First, each individual point is given a feature vector rather than the trajectory as a whole. The 4 features we used to describe each point in each trajectory are distance, velocity, angle, and angular velocity. Distance is the radius from the center of the image to the point in the path. Velocity is the distance (in pixels of the overhead image) that the point has traveled in the last time step. The angle is the direction in which the point moved from the last point in the last time step with respect to the direction of the asset ship's bearing. Angular velocity is the change in the angle from the last point over time since the last detection; for our synthetic data, the angular velocity is 1. We propose the transition of the features for a trajectory from point to point be a Markov process. With our training data, we build a transition matrix for each of our training samples that best describes how the features transition. For classification, any new trajectory is compared it to each of our models and find the likelihood of those feature transitions occurring given the model's transition matrix. We categorize the new trajectory based on which model had the highest likelihood of those feature transitions occurring. We found that the MTFC approach is less sensitive to variations in the parameters of the trajectory (i.e., sample frequency and number of points in the path), but it has a slow inference time which makes it impractical for use in real time.

C. Naive Bayesian Radial Classification

We propose to convert the pixel grid coordinates of each point of the detected boat into polar coordinates with the asset ship at the origin as a distance from the boat (radius) and its angle relative to the bearing of the ship (θ) from Figure!1. This approach to modeling indicates more information that is useful to us since these values are relative to the ship's

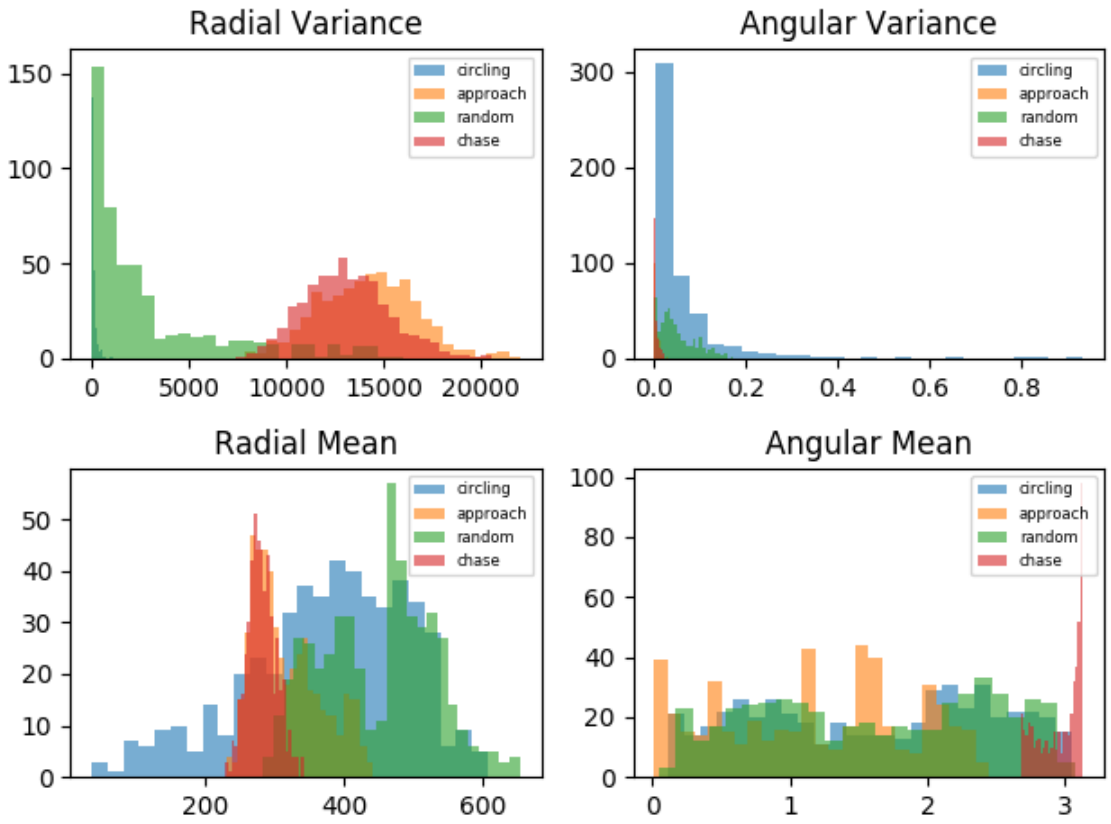

Fig. 4: Naive Bayesian Radial Classification (NBRC) feature distribution per different activity label.

position. θ is calculated by taking the inverse cosine of the dot product between the normalized vectors representing the ship's bearing and the direction to the tracked ship. This means θ will be between 0 and π as the absolute angle, as opposed to representing the point in standard polar coordinates with θ ranging from 0 to 2π. As we are not concerned on which side of the ship the activity is taking place, and symmetric tracks about the ship's bearing should be classified the same. The tracks modeled this way are robust within and different for different classification types. An approaching or chasing path will not have much variation to θ over the course of its path, as it approaching the ship at the same angle. The radius for the approaching activity will change quickly as it moves in. On the other hand, a circling trajectory will have a lot of variation in θ as it moves around the ship; however, the radius should not vary much. We propose to represent each trajectory as a feature vector of length 4 that contains the angular mean, angular variance, radial mean, and radial variance, as illustrated in Figure 4. The mean tell us roughly where on the overhead view the activity is taking place. This is especially important when differentiating between a track that is chasing, which is when the boat closing in from the rear, and a track that is approaching, which is closing in from any other direction. The variances tell us how much the track changes in direction and distance.

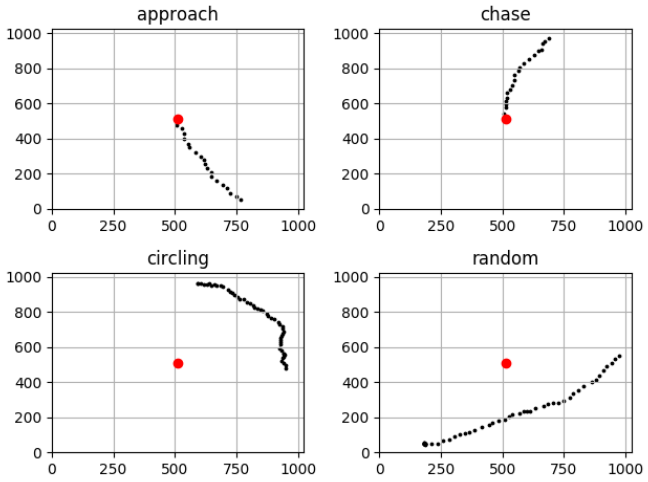

Fig. 5: Sample synthetic trajectories representing approach, chase, random, circling

IV. Data Synthesis From Real Data Examples

The IPATCH activity scenarios [5] provided us a blueprint on what defines a specific maritime vessel activity w.r.t. autonouis ship: appraching, circling, chasing another vessel, and random [5]. set which provides a set of videos from

different angles facing outward from cargo ships; these videos show simulated scenarios of small boats performing various maneuvers. Unfortunately, this data set does not contain enough footage to train any kind of model that can classify boat paths in the way we want. For this reason, we chose to create a synthetic data set of labeled trajectories to train and test our models.

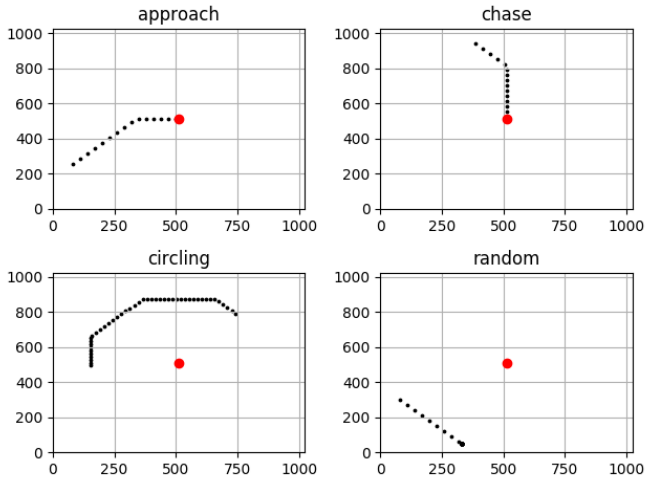

Fig. 6: Synthetic generated trajectories with no noise variation

Each trajectory in our synthetic data set is generated by one pixel move at a time. Our method of generating trajectories requires 3 parameters: the number of points, a sample frequency, and a probability distribution. When detecting tracks in the real world, the sample frequency will be the time between when each detected object is recorded and mapped to the overhead, but for our synthetic data, it is the number of pixel moves between each point. The probability distribution is given as a vector of length 9 whose values sum to 100. From a given point, there are 9 possible directions for movement: remaining in the current location or moving to any of the 8 surrounding pixels. To generate a trajectory, we determine a starting location and sort the possible directions we can go based on how well they will fit the trajectory we are trying to generate. Then we randomly select the direction to go based on the probability distribution provided. Therefore, the probability distribution should have higher values first that decrease so that our trajectory stochastically moves in the selected direction. This process is repeated with each move, and points are recorded based on the sample frequency until we have the desired number of points.

Approaching and chasing are the simplest trajectories to generate, as they are the paths that move towards the boat with some random noise as not to be perfectly straight lines, and they have a starting point somewhere on the edge of the overhead model domain. Approaching and chasing are only different in that chasing will come from behind, as we assume the boat is moving when the detection is made, while approaching can come from any other side.

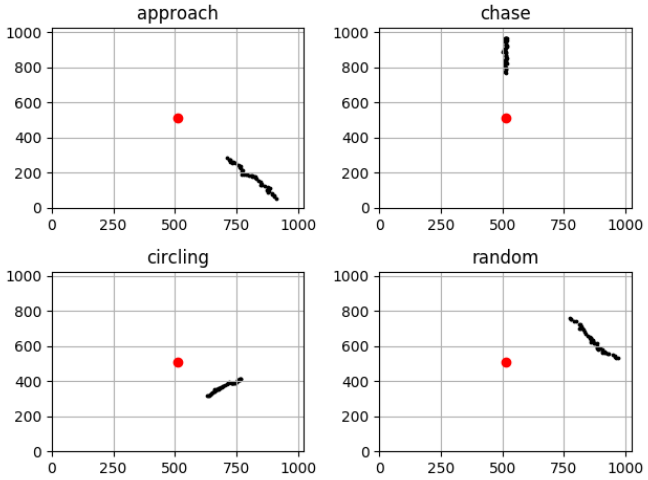

Fig. 7: Synthetic generated trajectories with high noise variations

Sample Frequency	35
Points	50
PD Training and Testing	20, 15, 15, 10, 10, 10, 10, 10, 0
PD Too Much Noise	20, 10, 10, 10, 10, 10, 10, 10, 10
PD No Noise	100, 0, 0, 0, 0, 0, 0, 0, 0

TABLE I: parameters for synthetic data sets

The random trajectory type is made in a similar way: we choose 2 random points on the overhead domain between which the path will move with the condition that the center, where the asset ship is located, is at least 150 pixels from the line between the 2 points so the path does not appear to be chasing or approaching. The circling trajectory type is slightly more complex to generate. It is a path that starts at a random location in the overhead domain that is at least 150 pixels away from the center and moves around the asset in a circular fashion. This will only end up being a partial circle, as it is unlikely that a ship will completely circle another ship in the real world. The most appropriate parameters for generating data were found through trial and error. We can see a real trajectory from the IPATCH data set in figure 1,as well as samples from the parameters we used to train our models, Figure 5 examples of trajectories with no noise Figure 6, and trajectories with too much noise Figure 7. The parameters for these synthetic paths can be found in Table I.

V. Experiments

The performance of the propose 3 models on synthetic dataset is illustrated in the form of the confusion matrices in Figure 8. Each row indicates the true label of a predicted sample, and the columns represent the number of times the model predicted that class. The results presented for our NBRC and MTFC approaches used a training set of 500 samples of each type for training and another set of 500 samples of each type for testing. Our Computing with Words approach does not need training data, so these results are using the same testing data as other two approaches. Note that the

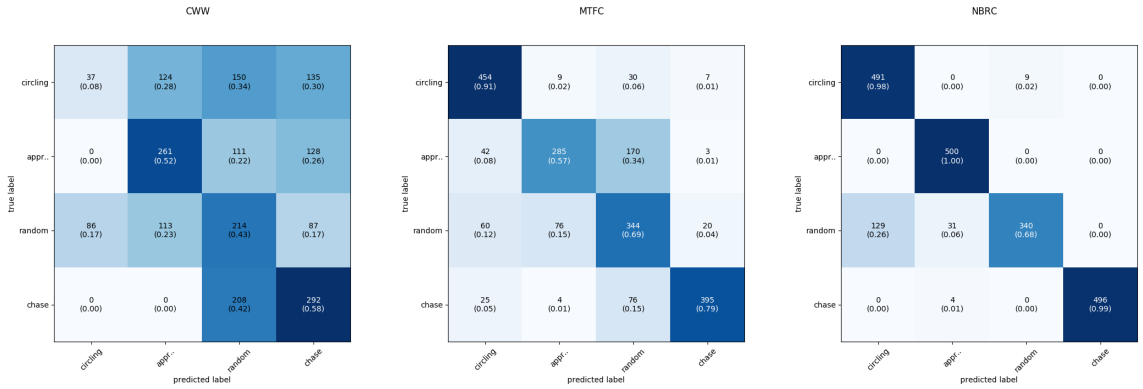

Fig. 8: (left) confusion Matrix for CWW, accuracy of the approach is 0.44; (center) confusion matrix for MTFC is 0.74; (right) confusion matrix for the NBRC is 0.91. Comparable accuracies are illustrated in Figure 9.

rows of the confusion matrix for this approach do not sum into the 500 samples used. With this model, a trajectory may not meet the conditions for any classification and remains as default. Although the correct identification of chase and approach trajectories may not be proficient, this model mostly struggles with identifying random and circling trajectories. Since circling trajectories are only partial circles, as we would not expect a boat to perform a full or large portion of a circle around the asset ship, it is not too often that it meets the condition of staying within the same region and hitting 3 of the quadrants. Moreover, if this model predicts circling, it is more likely that it is a random path than circling.

A. CWW: Computing With Words

The Computing with Words method was the most basic model we tested, and the results we got from this method are in the confusion matrix in Figure 8(left). Its overall accuracy is 0.4, which is better than the expected 0.25 if predictions were completely random. However, this accuracy is clearly insufficient and only serves as a base line. One of the main disadvantages of this method is that the sections for this model can be very large, especially when moving between quadrants in the same region. This means a large amount of movement may be needed for a new part of the string to be created. One workaround is to add more quadrants within each region since the region thickness can already be varied, but this would require new criteria for classification, and it will be ship dependant. We also find that the accuracy for CWW varies greatly with different parameters for our synthetic trajectories. In fact, some experiments with nosier trajectories yield worse than random guesses with the same classification criteria that provided decent results in our standard trajectories.

B. MTFC: Markov trajectory Feature Classifier

The results for the Markov trajectory feature classification in Figure 8(center) shows a better performance than CWW. The accuracy measure was 0.739, which is better than Computing with Words, but there is room for improvement. Although the results for this approach are better than the Computing with Words (CWW) approach, there is still a high level of inaccuracy, especially when considering the probability of the model being correct when predicting random trajectories. We speculate that this is due to the nature of the Markov process which involves making decisions based on how the features change from point to point rather than the values of the features themselves. There is information on the angle and distance of points in the paths relative to the asset, which are the differentiating factors between random trajectories and chase or approach trajectories; however, the way those features change from point to point seem to be very similar, causing the model to struggle. One problem that is not visible in the raw data is the inefficiency of this method causing it to be very time consuming. While it took seconds to run 2000 samples with the other 2 models, this method took 3.6 hours. The reason for this is the score function of hidden Markov models takes a long time to compute, and each of the 2000 test trajectories evaluated were scored against 2000 training models, each one taking about 6 seconds to compute. This makes this method infeasible for use in real time against real-world data. Ideally we would have a single model that defines how the features for any given trajectory class transition from point to point.

C. NBRC: Naïve Bayes Radial Classifier

Naïve Bayes Radial Classifier (NBRC) performance on the same dataset is captured in Figure 8(right). Each radial feature in Figure 4 roughly follows a Gaussian distribution. Once we learn the mean and standard deviation for each feature of each class, we can use that information to decide to which class a new trajectory belongs by finding which class has the highest likelihood given its feature values. In our implementation, we used the Scikit-learn Gaussian Naïve Bayes classifier trained on our synthetic data. Our Naïve Bayes Radial Classifier showed far more promising results than either of our baseline approaches with a total accuracy of 0.91, see Figure 9. There is a degree of similarity between this method and Computing with Words (CWW). A trajectory passing through multiple quadrants corresponds to a high angular variance and passing through multiple regions corresponds with a high radial variance. In terms of quantifying these metrics of the trajectory, it is much more efficient to numerically calculate these variances

for a given trajectory and apply Bayes' theorem to find the conditional probability that it belongs to each class given those values for other samples. One notable disadvantage of this method is the lack of temporal information used. This would be an issue if we want to differentiate between paths moving towards and away from the asset ship. Moreover, limitation of path types that can be classified since all types must be separable by the angular mean, radial mean, and variance. However, complex paths do not appear frequently and in the context of threat detection; therefore, the recognition of such paths may be unnecessary, and the classification of more basic path types is more useful.

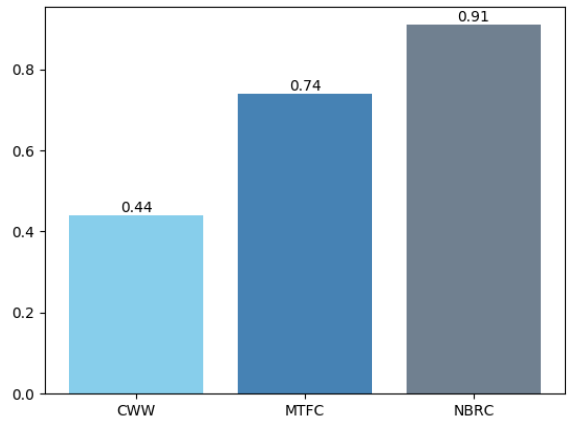

Fig. 9: three model comparison in terms of accuracy on the same synthetic dataset: Naïve Bayes classifier on Radial features exhibits dominant performance.

VI. Conclusion

This paper describes our work on activity classification of small maritime objects on an overhead-view from multiple visual feeds. Our NBRC approach, which involves using a Gaussian Naïve Bayes classifier over the radial and angular variance and means of trajectories showed far better results than our 2 baseline approaches which we compared against as illustrated in Figure 9. NBRC approach achieves 91% accuracy on test data. We conculde that the use of radial features from overhead multi-camera projects leadfas to most reliable results in maritime vessel classification. Future research focuses on contextual threat analysis that is building on activity recognition for autonomous ship camera systems.

VII. Acknowledgments

This material is based upon work supported partly by NSF Research Experiences for Undergraduates in Smart and Connected Communities under contract 1757893 and NAVAIR under contracts STTR N68335-16-C-0028 and SBIR N68335-18-C-0199. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

We would like to thank all current and former students in Data Lab @ TXST [12] and all participants of NSF REU SCC [13] that have helped with suggestions, coding or asking great questions to shape this paper. We want to extend our special thank you to Alan Turner as he implemented the first version of multi-camera object tracking and overhead plane projection [9] and to Diego Sebastian Pelayo-Santana for annotating vessel activities in IPATCH [4] data, training and evaluating more efficient vessel tracking deep learning models, and for publishing the code and findings [9].

References

[1] A. Octavian and W. Jatmiko, "Designing intelligent coastal surveillance based on big maritime data," in *2020 International Workshop on Big Data and Information Security (IWBIS)*, 2020, pp. 1–8.

[2] D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, and M. Fiorini, "Enhancing automatic maritime surveillance systems with visual information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 824–833, 2017.

[3] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J. M. V. García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Z. H. Bhuiyan, S. Särkkä, and V. V. Lehtola, "Sensors and ai techniques for situational awareness in autonomous ships: A review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2020.

[4] L. Patino, T. Nawaz, T. Cane, and J. Ferryman, "Pets 2017: Dataset and challenge," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 2126–2132.

[5] M. Andersson, R. Johansson, K.-G. Stenborg, R. Forsgren, T. Cane, G. Taberski, L. Patino, and J. Ferryman, "The ipatch system for maritime surveillance and piracy threat classification," in *2016 European Intelligence and Security Informatics Conference (EISIC)*, 2016, pp. 200–200.

[6] D. Qiao, G. Liu, T. Lv, W. Li, and J. Zhang, "Marine vision-based situational awareness using discriminative deep learning: A survey," *Journal of Marine Science and Engineering*, vol. 9, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/4/397

[7] N. Warren, B. Garrard, E. Staudt, and J. Tešić, "Transfer learning of deep neural networks for visual collaborative maritime asset identification," in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, Oct 2018, pp. 246–255.

[8] D. Heyse, N. Warren, and J. Tešić, "Identifying maritime vessels at multiple levels of descriptions using deep features," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, T. Pham, Ed., vol. 11006, International Society for Optics and Photonics. SPIE, 2019, pp. 423 – 431. [Online]. Available: https://doi.org/10.1117/12.2519248

[9] D. S. Pelayo-Santana and A. Turner, "Overhead object tracking to activity recognition," https://github.com/DataLab12/trackActivity, 2020.

[10] M. N. Kurt, Y. Yılmaz, and X. Wang, "Real-time nonparametric anomaly detection in high-dimensional settings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2463–2479, 2021.

[11] J. Tešić, D. Tamir, S. Neumann, N. Rishe, and A. Kandel, "Computing with words in maritime piracy and attack detection systems," in *Augmented Cognition. Human Cognition and Behavior*, D. D. Schmorrow and C. M. Fidopiastis, Eds. Cham: Springer International Publishing, 2020, pp. 434–444.

[12] J. Tešić, "Data lab @ txst," DataLab12.github.io, 2021.

[13] V. Metsis, "Nsf reu site: Research experiences for undergraduates in smart and connected communities (reu scc)," 2019.