

Maritime SAR with Computer Vision

Shaurya Sen¹, Logan deLaar¹, and Michael Novitzky²

Abstract—This paper presents the development and implementation of autonomous maritime search and rescue (SAR) operations, integrating Computer Vision and the MOOS-IvP autonomy stack [1], [2]. The research project aims to improve the efficiency of maritime Search and Rescue operations using unmanned surface vehicles (USVs). By incorporating YOLO (You Only Look Once) for real-time object detection and developing a homography algorithm for spatial analysis, the project provides a robust solution for locating a Man Over Board (MOB) in marine environments. The MOOS-IvP framework serves as the backbone of the system, allowing for control of autonomous behaviors in the marine robots. MOOS-IvP operates through communication between MOOS apps, and for this project, the use of two novel MOOS applications: one to leverage YOLO to detect persons overboard in the camera’s video feed, and another for location calculation, using the homography algorithm to transform image coordinates into real-world locations. The custom model was trained with thousands of data points, enabling the marine robots to reliably detect an MOB. Additionally, a path finding application was developed to guide the USV in taking the optimal path to survey a marine environment. The results of this research can be enhanced in the future with improvements to the model, incorporating additional sensor data like LIDAR or infrared, and deploying multiple USVs, for collaborative SAR operations. This project marks a significant advancement in the use of autonomous systems and CV in maritime SAR efforts.

I. INTRODUCTION

Maritime search and rescue (SAR) operations are critical for saving lives in aquatic environments but face challenges as they are today. The vastness of search area in marine landscapes, and dangerous weather conditions, make SAR missions both inefficient and dangerous for those involved. These challenges with the conventional approach, which relies heavily on human teams, put both the rescuers and MOB at greater risk. This paper addresses these challenges by fully automating the search and identification of MOBs, allowing for safer and more efficient SAR missions. This research focuses on developing and implementing autonomous technology in order to enhance maritime SAR efficiency. By using unmanned surface vehicles (USVs) equipped with the ability to survey bodies of water and autonomously locate MOBs, these missions have been made more safe and efficient. The purpose of this paper is to detail the development of a SAR system that uses the MOOS-IvP

autonomy stack on a SeaRobotics Surveyor M1.8 maritime robot [3] combined with computer vision to make the missions more effective and to mitigate risks to human life. The introduction of USVs in SAR operation is a step that will increase operational hours, reduce human risk, and achieve more efficiency, presenting an innovative approach to a long-standing problem.

SAR With CV team analysis objective:

•**Motivation and Context:** Human team SAR operations have several limitations such as lack of efficiency and safety, whereas USVs allow for more constant surveillance, and more efficient detection while minimizing safety risks.

•**Analysis Objective:** The objective is to create a maritime SAR solution automated with USVs. By integrating CV with the MOOS-IvP framework, these robots should be able to survey maritime environments and autonomously locate MOBs. This research aims to demonstrate how computer vision can be used with USVs to make maritime SAR missions more effective.

•**Typical Behavior:** The USV will start off by following a predetermined search pattern, generated by a pathfinding application specific to the marine environment. During this time, the YOLO MOOS app will constantly analyze the video stream from the robot camera to determine if a man is overboard. Upon detecting a MOB, the location calculation MOOS app employs the homography algorithm to estimate the MOB’s relative distance to the robot. Using the robot’s GPS coordinates and the MOB’s relative distance, the exact coordinates of the MOB are determined and reported to shoreside where a rescue response will be initiated.

II. METHOD

A. MOOS-IvP

The autonomous SAR operations are powered using the MOOS-IvP autonomy stack. MOOS-IvP is an open-source framework that allows developers to define the behavior of autonomous systems. MOOS-IvP is operated through the communication between a network of MOOS processes also known as MOOS apps. MOOS-IvP already has a wide range of ready-to-go applications to simulate and run operations on marine vehicles. One example of an integral MOOS App is the IvP Helm, an autonomy engine that allows developers to define behavior files to control decision-making and navigation of the autonomous system. By using the MOOS middleware, IvP Helm can interpret these behavior files and transform it into instructions for marine vehicles. For this project, the modularity offered by MOOS-IvP was leveraged to create two new applications, making the USVs applicable in SAR operation contexts. The purpose of these

*This work was not supported by any organization

¹Shaurya Sen Logan deLaar and are Research Interns with the Robotics Research Center, United States Military Academy, West Point, NY 10996, USA shauryasen12@gmail.com, delaarlogan@gmail.com

² Michael Novitzky is Faculty with the Robotics Research Center and the Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY 10996, USA michael.novitzky@westpoint.edu

applications are to serve as an interface to the robot's hardware, enabling the YOLO and homography algorithms to use the robot's camera sensor and GPS coordinates. These are called the Object Detection app and the Location Calculation app. The Object detection app uses YOLO, a fast object detection algorithm to constantly scan frames from the video stream taken from the robot's onboard camera to detect if there is a man overboard. The Location Calculation app uses a custom-built homography algorithm in order to transform a detected MOB's image coordinates into an actual geographical position using the USV's GPS data.

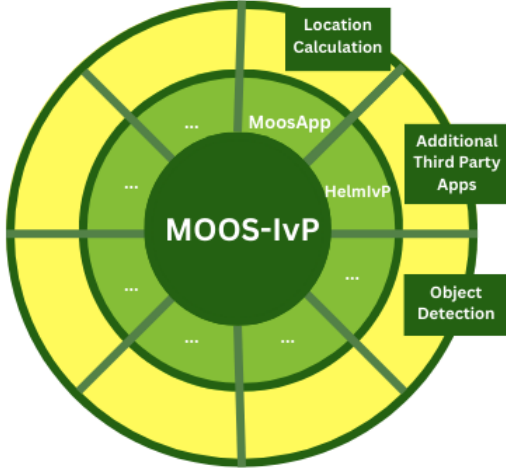


Fig. 1. MOOS-IvP Community Diagram

Figure 1 shows a depiction of the MOOS-IvP community. The center of the wheel represents the MOOS-IvP Core. This includes the MOOSDB which facilitates the communication between MOOS Apps, and the scheduling algorithms. The inner darker wedges represent the read-to-go publicly available applications that provide basic functionality, this includes the IvP Helm mentioned earlier. The outer wedge represents the additional MOOS applications and wrappers that can be created by third-party developers. This is the category that the Location Calculation and Object Detection apps fall under. This modularity enables effective code reuse, for example, the Object Detection app can now easily be used in other contexts outside of maritime SAR operations. These independent applications communicate with each other using a publish/subscribe architecture. In this structure, an app can publish data to, or subscribe to data from, the MOOS Database (MOOSDB). This architecture makes the MOOS community more modular, and easy to add or replace apps.

B. Object Detection

In the context of maritime SAR operations, the object detection MOOS app is used to facilitate the usage of the robot's camera in order to detect MOBs. As the USV surveys the area, the camera captures a continuous stream of data which the application parses frame by frame, using the YOLO (You Only Look Once) algorithm to detect persons overboard. This process ensures constant surveillance, one of

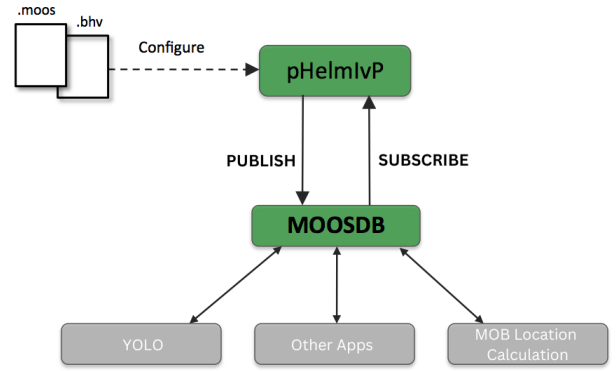


Fig. 2. Publish Subscribe Architecture

the benefits of using autonomous systems for SAR. If a MOB is detected, the app publishes the pixel coordinates of the MOB within the video stream image to the MOOSDB. The Location Calculation App will then retrieve this information and use it to calculate the relative position of the MOB. The Object Detection app does not subscribe to any data from the MOOSDB and is something that can be constantly kept running with no external inputs needed. Instead, it only focuses on its important publishing role in the MOOS-IvP community. Upon detecting a MOB, it publishes several key pieces of information regarding the MOB to the MOOS Database. It publishes the pixel coordinates of the detected MOB within the video frame. The YOLO algorithm draws a bounding box around detected objects, and the application takes this and publishes the coordinates of it which are later used by the location calculation process to calculate the MOB's relative distance to the USV. It also constantly publishes a boolean which indicates whether a MOB has been detected or not, and when a MOB is detected, this boolean is set to true to indicate to shoreside that there is a MOB detected. In addition, it also publishes the confidence level of the detection and the name of the object that is detected. In the case of SAR operations, the name would be a MOB, however, when testing life vests and other objects were also used.

1) *YOLOv5*: The algorithm of choice for object detection for the SAR operations was YOLOv5. YOLO was chosen because it has proven to be both fast and accurate which is necessary for the real-time object detection that is required. It is a state-of-the-art CNN that allows for multi-class detection which allows the USV to survey an area for a wider range of objects, including other sea robots in order to prevent collisions and to differentiate between humans in water vs humans on land. This algorithm was tailored to be useful in SAR scenarios by training a model with thousands of images, of both humans in water, and life vests in water which was used for the demo.

2) *Further Uses of Object Detection*: The Object Detection app was designed with the modularity of the MOOS community and the versatile use of object detection in mind. Object detection can easily be used in applications

of autonomous vehicles far outside the realm of maritime SAR and with this context it was decided to make the moos app as generalized as possible. It can be used, not only with a model trained on detecting MOB's but any custom model. As long as the autonomous system features a camera, and a model, this moos app can be used to facilitate real-time object detection. With this in mind, the goal is to explore how object detection can be applied to other autonomous vehicles, such as providing aerial SAR support.

C. Location Calculation

The location calculation MOOS app is responsible for determining the precise location of a detected person overboard. As the robot is surveying a marine environment, this app is not in use. However when a MOB is detected, the location calculation app uses the pixel coordinates that the YOLO app publishes and upon receiving them, the homography script is used to calculate the relative x and y coordinates of the MOB. By combining this information with the robot's current position and heading, the app determines the precise geographical location of the MOB. As a part of the MOOS community, the location calculation app subscribes to the x, y, and heading of the robot, and the pixel coordinates of any detected person overboard. Using this information, it calculates and publishes the MOB's relative distance to the robot, and the MOB's precise location.

1) *Homography*: The homography algorithm transforms pixel coordinates in an image into relative geospatial coordinates. This is done using a homography matrix, which maps pixels in the video stream with predefined real-world points taken during calibration. OpenCV was used to process images for calibration and detecting chessboard corners to establish real-world reference points. It performs camera calibration using the detected points to determine the camera matrix and distortion coefficients. The calculated calibration parameters are stored for future use, providing consistent imaging metrics across different sessions. By setting up a camera at the same height from the floor as the robot's camera would be from the water, an accurate and consistent homography matrix tailored to the USV was obtained.

D. Path Planning Algorithm

The path planning algorithm begins by defining the search area as a polygon based on user-selected GPS coordinates, which is then subdivided into a grid aligned with the vehicle's sensor range (10 meters in this case), ensuring thorough coverage. Each grid cell represents a segment of the search area that the Unmanned Surface Vehicle (USV) will survey, with waypoints calculated at the center of these cells to form a systematic lawnmower pattern. This pattern is optimized to minimize overlap and ensure efficient coverage by arranging waypoints in an alternating sequence, allowing the USV to move in a back-and-forth motion across the search area.

E. MOB Detection Model

1) *Data Collection*: In order to have a good model a large and diverse dataset was needed. In order to achieve

this, a script on the USV's onboard Raspberry Pi was run to automatically capture images at random intervals. The USV was then taken out onto Lake Popolopen to gather data. By using the camera on the USV to collect training data, the model was optimized for the specific conditions of the USV. Data collection was conducted on multiple days to capture variations in time of day and weather conditions. By the end of it, over 8.5k images that were usable for training. In a dataset of 8,607 images, MOB's, people on land, boats, life vests, and other robots were manually annotated.

2) *Model Training*: Roboflow was used for the management of the model training process and the organization of the data. After the data was collected and annotated, it was trained a YOLO model splitting up the images to have 10 percent reserved for validation in order to assess the accuracy of the model. The training process was iterative, with new models being created based on findings from previous ones and as additional data was acquired. One example in an early model was the wide-angle lens of the robot's camera causing the model to identify visible parts of the same robot as a different robot. However, after several iterations in the model, a model that was accurate and reliable was developed. To more rigorously test the model it was deployed on the robot and brought it to the lake to ensure it was properly detecting each object.

III. RESULTS AND TESTING

As iterations were made through various models, the performance improved. By the final iteration, a robust and reliable MOB detection model for SAR operations had been created. For testing and demonstrations, life vests were primarily used in the water, so the model was also trained to detect those. During testing, the model proved to be accurate.



Fig. 3. Life Vest Detection

The endeavor to create an autonomous maritime SAR with CV culminated in successful testing. First the USV navigated to the middle of Lake Popolopen to initiate the SAR procedure. Once in position, the USV began to autonomously

survey the area, using the path planning algorithm. This pattern was generated by the pathfinding application. After drawing a polygon around the area that is to be surveyed, the pathfinding app calculates the an optimal path such that all portions of the area are covered by the USV's line of sight. The USV then follows this path back and forth. As the USV is executing this search pattern, the onboard cameras and MOB detection model are put to use to constantly search the video feed for MOBs. Upon detecting a MOB, or for the purposes of this demonstration a life vest, the USV stops searching and stays in place. It then calculates the precise geographical coordinates of the object and sends it to shoreside. This will appear on the shoreside as a pin on their map, giving them the exact location of the MOB so a rescue response can be initiated. Through repeated tests, the accuracy of the model and the reliability of the autonomous SAR process were proven. Overall this project offers a robust and reliable solution to autonomous maritime SAR operations.

IV. APPLICATIONS

The applications of Computer Vision in maritime SAR operations present opportunities for improvement and have room for applications beyond the tests so far. The successful tests show that CV can be effectively paired with USVs, paving the way for expanding uses in maritime activities. The primary application of this technology is expanding SAR operations. The USV's ability to autonomously search large bodies of water, identify MOBs, and communicate their location to rescue teams will allow for more efficient response times and more persistent surveillance in areas where they may be needed. This can be expanded to using and coordinating multiple USVs to cover larger areas and increase efficiency. By incorporating more sensor data like LIDAR or infrared, the upkeep and efficiency can be increased, allowing it to be operational where visibility is low or during the night. The same technologies developed in this project can be implemented in other autonomous systems as well including UAVs. With UAVs, greater efficiency in searching can be achieved.

REFERENCES

- [1] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with moos-ivp," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.
- [2] www.moos-ivp.org, "The MOOS-IvP Home Page," On line <https://oceanai.mit.edu/moos-ivp/pmwiki/pmwiki.php?n=Main.HomePage>, (Accessed: 1 April 2023).
- [3] SeaRobotics, "SeaRobotics Surveyor M1.8," On line <https://www.searobotics.com/products/autonomous-surface-vehicles/sr-surveyor-class>, (Accessed: 20 August 2024).