

Image Processing and Model-Based Spill Coverage Path Planning for Unmanned Surface Vehicles

Shaocheng Luo*, Yogang Singh*, Hanyao Yang*, Jun Han Bae*, J. Eric Dietz†, Xiumin Diao‡, and Byung-Cheol Min*

*SMART Lab, Department of Computer and Information Technology, Purdue University, IN 47907, USA

†Department of Computer and Information Technology, Purdue University, IN 47907, USA

‡School of Engineering Technology, Purdue University, IN 47907, USA

Abstract—Remote sensing technology and unmanned surface vehicles (USVs) have great potential for spill coverage applications. However, there is still a lack of research on adequate path planners for USVs based on remote sensing images. In this study, we propose an image processing and model-based path planner that can generate an efficient path for a USV to cover a spill according to aerial images from remote sensing. We mathematically formulate a spill processing model of the conceptual USV, which can remove the spill by suction mechanism, to determine its speed limit and operation range. To perform coverage with high completeness to the spill in the workspace, we first develop an image segmentation strategy and partition the area with geometric tessellation. We then formulate the coverage problem with tessellation as a traveling salesman problem (TSP) and utilize the self-organizing map (SOM) approach for effective path planning. Finally, with real aerial images containing spills, we demonstrate the effectiveness of the proposed path planning method.

Keywords—Unmanned surface vehicles, Spill removal, Image processing, Remote sensing, Geometric tessellation, Traveling salesman problem, Self-organizing map

I. INTRODUCTION

Harmful spills, such as spill blooms and chemical leakage, in aquatic environments, including ditches and reservoirs significantly affect ecosystems and public health. Timely response and emergent restore operation are necessary to prevent further damages. However, the lack of trained labors and potential dangers for human operators in hazardous environments become constraints and usually delay the process.

Water surface robots such as unmanned surface vehicles (USVs) have been studied extensively recently. With aerial images provided by manned/unmanned aerial vehicles (UAVs) or satellites, the USV can be used in wide applications including riverine environmental monitoring [1], flooded open-pit mine 3D reconstruction [2], ocean cleaning [3], marine incident response [4], and water and sediment sampling [5].

Nevertheless, a spill cleaning solution based on remote sensing and USV technologies has not yet been fully explored, mostly because of the lack of a functional spill removal USV and an adequate planner that enables practical and efficient USV operation with aerial images. For instance, [6] and [7] implemented a monocat and catamaran type of water surface vehicles, respectively, to confront oil spills. However, their traditional conveyor belt based appliance can hardly separate oil spills from water, which hence hinder a further promotion of efficiency. Moreover, most of the vehicles used in large-scale removal operations, such as proliferated plankton spill

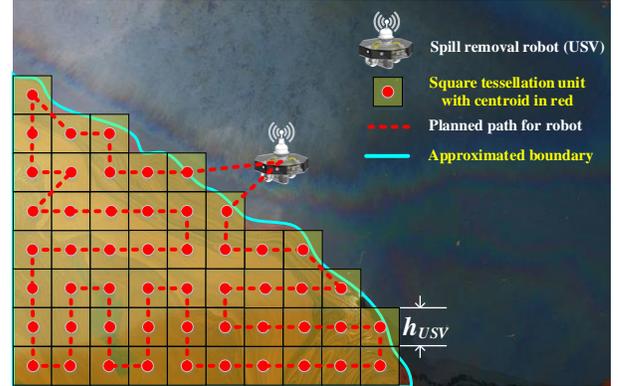


Fig. 1. A concept of the proposed image processing and model-based spill coverage path planning for a USV. Provided that an aerial image is obtained with remote sensing techniques, an effective path is planned using our proposed image processing and geometric tessellation strategies for the USV to travel and remove the spill. The size of square tessellation block h_{USV} can vary and is decided by the spill removal rate of the USV.

processing and oceanic oil leakage restore, are designed of being human-operated. No global information is utilized to enable autonomy. In practice, the boundary of the spill can be extremely coarse due to its own molecular characteristics and interaction with water, which presents a substantial challenge for the realization of a complete coverage operation. A practical solution for coverage control in the presence of coarse boundary is rarely researched.

Therefore, in this paper, we propose an effective path planner that can generate an efficient path for a USV to cover a spill with aerial images provided from remote sensing. To detect the spill area to be covered from an image, spill boundary extraction and approximation, and square tessellation are achieved via image processing. The model and spill process capability of the conceptual USV platform are considered to develop an effective path planner. The concept of the proposed path planner is depicted in Fig. 1, where an efficient path is generated from the aerial spill image.

The rest of this paper is organized as follows. The research problem and approach are briefed in Sec. II. In Sec. III, we mathematically formulate the process rate of spill absorbed to the USV with suction while the USV is moving. Based on the bounded spill process capacity, we determine the maximum USV traveling speed v_{max} while separating the absorbed spill from water. In Sec. IV, we introduce the developed image segmentation strategy and implement a geometric tessellation

with squares over the area of interest where the spill locates at. The size of squares for tessellation is decided according to the USV process limit. After formulating the tessellation coverage operation as a traveling salesman problem (TSP), the suboptimal path planned for the USV is generated by applying the self-organizing map (SOM) approach in Sec. V. The path planning results are presented in Sec. VI with diverse aerial images containing ground truth. The paper is concluded in Sec. VII.

II. PROBLEM STATEMENT AND PROPOSED APPROACH

In this research, we desire to realize a path planner that takes aerial images from remote sensing as input and generate an efficient path for USV to travel and remove the spill. To do so, the following two questions must be addressed:

- Provided a generic USV that can remove the spill by suction, what the maximum traveling speed v_{max} and operation range h_{USV} are, assuming that the USV has a bounded spill process capability denoted as \mathcal{V} .
- Provided that the USV is traveling at the speed of v_{max} with an operation range h_{USV} in removing the spill, how to generate the shortest path for it to expedite the operation while maintaining a high coverage rate to the spill in the workspace.

For the first question, we propose a mathematical model that takes into account how the spill and the USV mutually affect each other when the USV is moving. Here, the amount of absorbed spill is calculated by its area. Provided a bounded capacity of \mathcal{V} , we are able to determine the maximum USV traveling speed v_{max} . After analysis and demonstration, we justify that the higher speed the USV maneuvers, the less operation range h_{USV} is. The detailed reasoning can be found in Sec. III.

Define the operation range h_{USV} for USV traveling at speed v_{max} , we address the second question by partitioning the workspace using geometric tessellation method, in which the dimension of tessellation blocks is bounded by h_{USV} . Since we do not want the USV to traverse the entire workspace and waste time, we utilize image processing techniques beforehand and only do tessellation to the spill rather than the entire workspace. Due to this, the spill coverage problem is formulated as a TSP, and an efficient path that the USV follows to visit each tessellation block needs to be determined.

Because of the NP-hard characteristic of the TSP, we use the SOM approach, along with the greedy algorithm and 2-Opt algorithm as comparisons, to produce a suboptimal solution. The path derived is efficient in the sense of time consumption, because the USV travels at the maximum speed along the shortest path while removing the spill. We also show that the geometric tessellation can deal with coarse spill boundary cases, which are challenging in practice yet sufficiently explored. Our solution is conceptually depicted in Fig. 1.

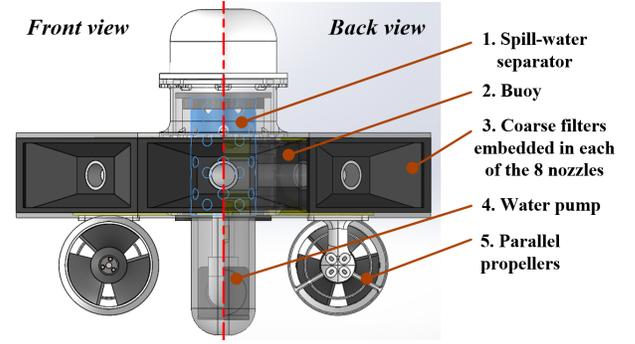


Fig. 2. Conceptual design of the spill removal USV. The USV uses a water pump to create suction and absorb spills coming from the eight nozzles with both coarse filters and a separator.

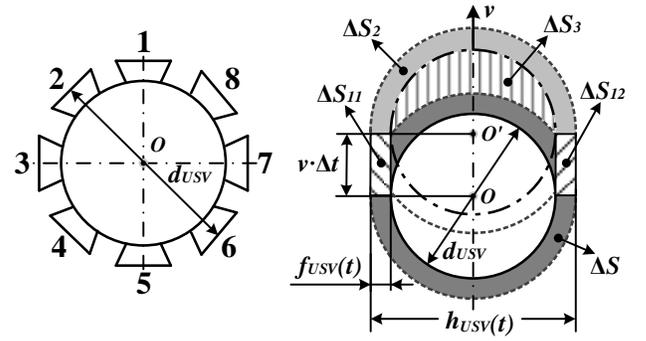


Fig. 3. The nozzle layout of a USV. The left figure shows the indices of nozzles, while the right one shows the spill collected (shaded areas) when this robot is maneuvering with a velocity v .

III. SPILL PROCESSING MODEL WITH A USV PROTOTYPE

In order to show our contribution in determining the moving speed and operation range of the USV, the prototype of a generic USV needs to be demonstrated. We adopt a common two-propeller driven USV as the end effector for our developed planner, and we conceptually develop a polygonal shape USV platform equipped with eight evenly distributed vacuum nozzles along its edge, as shown in Fig. 2. This conceptual platform is inspired by our previous work [8], [9]. When the water pump runs and causes greater fluid pressure at the bottom of the robot than on the surface of the water, the spill on the surface can be sucked into the separator inside the robot for segregation.

A proper processing model is indispensable in determining the maximum speed for a USV, according to capacity. As shown in the prototype of USV in Fig. 2 and Fig. 3 (left), the diameter designed for USV is $d_{USV} = 350$ mm and eight nozzles are arranged along the robot body to enable spill removal operation.

When a USV is collecting spill, it should not exceed the maximum power of the water pump creating vacuum pressure and separator volume sufficient for separating spill from water. Obviously, a larger USV has the higher capability in spill removal. However, the size of USV depends on many factors, including waterproofing and security of other parts, and is therefore restricted. Because of the size restriction, too much

spill absorbed at a time may block the nozzles or damage the separator. Therefore, the maximum linear velocity will be calculated under the maximum capacity of the robot, which is denoted as \mathcal{V} in this paper.

In practice, not every nozzle has a chance to collect spill in processing. Hence, we introduce a coefficient γ_{nz} that represents the ratio of operating nozzles n_{nz} to the total number, i.e., $\gamma_{nz} = n_{nz}/8$. It is evident to see the spill collected by an individual USV can be formulated as $\gamma_{nz}\mathcal{V}$ in unit time. When a USV is static and has velocity $v = 0$, the width $f_{USV}(t)$ of the coverage area has to follow a static model shown in Theorem 1, which is depicted with a dark shadowed ring ΔS in Fig. 3 (right).

Theorem 1 (Spill Processing Model in Static). *If the USV stays still in place ($v = 0$) and is fully surrounded by the spill, the $f_{USV}(t)$ follows the conclusion below:*

$$f_{USV}(t) = \sqrt{\frac{\mathcal{V}}{\pi}t + c_1} - c_0, \forall t \geq 0 \quad (1)$$

where $c_0 = \frac{d_{USV}}{2}$ and $c_1 = c_0^2$. t refers to the elapsed time.

However, since the USV is moving dynamically with a velocity of v , an additional spill can spontaneously rush into the robot, in addition to the spill collected with vacuum water pump, creating an extra burden for the separator. Thus, we need a dynamic spill processing model to determine the optimal combination of spill absorbing rate and moving velocity v . We assume that when the robot is moving forward, it cannot capture spill located behind, as spill travels slowly. Thus, nozzles $\{1,2,3,7,8\}$ shown in Fig. 3 (left) are doing most of the collection work, meaning $\gamma_{nz} = 0.625$. Considering those facts along with Theorem 1, we build the dynamic processing model as below.

Theorem 2 (Spill Processing Model in Moving). *If the USV moves with a velocity $v \neq 0$ in a fully spill surrounded area, the spill collected with respect to the capacity of a USV has to satisfy the following equation set:*

$$\sum_i \Delta S_i = \Delta S_{11} + \Delta S_{12} + \Delta S_2 + \Delta S_3 = \gamma_{nz}\mathcal{V} \cdot t = 0.625\mathcal{V} \cdot t, \quad (2)$$

where

$$\begin{aligned} \Delta S_{11} &= \Delta S_{12} = v(t) \cdot t \cdot f_{USV}(t), \\ \Delta S_2 &= \frac{\pi}{2} [(f_{USV}(t) + c_0)^2 - c_0^2], \\ \Delta S_3 &= \begin{cases} c_0 v(t) \cdot t \sin c_2 + (\pi - c_2)c_0^2 - c_3(f_{USV}(t) + c_0)^2, & \text{if } v(t)t \leq 2c_0 + f_{USV}(t), \\ \approx 2c_0 v(t)t, & \text{otherwise.} \end{cases} \end{aligned} \quad (3)$$

$$\begin{aligned} c_0 &= \frac{d_{USV}}{2}, \\ c_2 &= \arccos\left(\frac{c_0^2 + (t \cdot v(t))^2 - (f_{USV}(t) + c_0)^2}{2c_0 t \cdot v(t)}\right), \\ c_3 &= \arccos\left(\frac{(f_{USV}(t) + c_0)^2 + (t \cdot v(t))^2 - c_0^2}{2(f_{USV}(t) + c_0)^2 t \cdot v(t)}\right). \end{aligned}$$

From Theorem 2, we can see the relationship between robot moving velocity v and spill coverage radius f_{USV} that lead to the maximum amount of spill harvested. In order to reduce the computation cost and simplify the further analysis, we seek for an approximate dynamic spill processing model that shows this relationship in a more straightforward way.

Lemma 1. *The relationship between robot moving velocity v and spill area coverage radius f_{USV} can be approximated as*

$$0.625\mathcal{V} = \begin{cases} 2vf_{USV} + c_4v + c_5\mathcal{V}, & \text{if } v(t)t \leq 2c_0 + f_{USV}, \\ 2vf_{USV} + 2c_0v + 0.5\mathcal{V}, & \text{otherwise,} \end{cases} \quad (4)$$

where $c_4 = c_0 \sin c_2$, and $c_5 = \frac{1}{2} - \frac{c_3}{\pi}$.

Proof. We examine the Theorem 2 and can show the following conclusions with ease:

$$\begin{aligned} 0 &\leq c_2, c_3 \leq \pi; \\ \Delta S_{11}, \Delta S_{12} &= \mathcal{O}(t^{1.5}); \Delta S_2 = \mathcal{O}(t); \Delta S_3 = \mathcal{O}(t), \end{aligned}$$

where $\mathcal{O}(\cdot)$ means asymptotic upper bound.

As time elapses ($t \rightarrow \infty$), we only remain those terms ΔS_x or part of a term ΔS_x with a higher order of t , thus (2) becomes

$$\begin{aligned} \sum_i \Delta S_i &= 0.625\mathcal{V} \cdot t \\ &= \begin{cases} 2vtf_{USV} + (\frac{\pi}{2} - c_3)f_{USV}^2 + c_0vt \sin c_2, & \text{if } v(t)t \leq 2c_0 + f_{USV}(t), \\ 2vtf_{USV} + \frac{\pi}{2}f_{USV}^2 + 2c_0vt, & \text{otherwise.} \end{cases} \end{aligned}$$

Eliminate the factor t from both sides of the second equal sign and let $c_4 = c_0 \sin c_2$, $c_5 = \frac{1}{2} - \frac{c_3}{\pi}$, we show (4). This concludes Lemma 1. \square

From (2) and (4), we can see that the faster the USV moves forward, the more spill is collected passively, meaning more spill are fed into the robot while less spill are collected with suction. By contrast, if the robot moves slowly, the fewer spill is flooded into the robot, but more of them are collected actively by the pumps. As a piece of evidence, the spill absorbing border f_{USV} grows further apart in a period of time when the robot moves slowly.

As we concluded earlier that the robot could not exceed a maximum speed of v_{max} because excessive spill collected may damage the robot and cause the task to fail. Such speed limitation v_{max} can be determined from the result of Lemma 1. It is preferable for the USV to move at as a high speed as possible when processing because this results in a shorter operation time. As the speed is bounded by v_{max} , we want the robot to maneuver at the maximum speed v_{max} without detected risks.

Lemma 2. The maximum speed v_{max} allowed for a USV processing within a non-hollow and symmetrical spill is

$$v_{max} = \frac{1}{16c_0} \mathcal{V}. \quad (5)$$

Proof. Considering (4), since the f_{USV} represents the spill actively collected by a USV, the faster USV moves, the less f_{USV} value is. Then we can simply let $f_{USV} \rightarrow 0$, thus we obtain \tilde{v}_{max} as below:

$$\tilde{v}_{max} = \begin{cases} \frac{5-5c_5}{8c_4} \mathcal{V}, & \text{if } \tilde{v}(t) \leq 2c_0, \\ \frac{1}{16c_0} \mathcal{V}, & \text{otherwise.} \end{cases}$$

We then decide $\frac{1}{16c_0} \mathcal{V}$ to be the candidate of v_{max} because it is derived on the basis that a robot moves with a higher speed, according to (3) and (4). This concludes the proof. \square

With a USV traveling at the maximum speed v_{max} , the corresponding f_{USV}^* can be obtained from (4), which will be used to determine the maximum size of a packing shape in workspace tessellation. However, if the real maximum speed of the USV is bounded mechanically and less than the computed v_{max} , the actual f_{USV} value may be greater than the computed f_{USV}^* . Ultimately, the width that the USV covers in maneuvering is $h_{USV} = 2f_{USV} + d_{USV}$. Our solution shows strong adaptation to diverse h_{USV} value, as demonstrated in Sec. VI.

IV. IMAGE SEGMENTATION AND WORKSPACE TESSELLATION

In this work, we present an image processing strategy that performs image segmentation with aerial images provided by remote sensing and geometric tessellation to the area of interest. Compared with other work done in spill removal operation, such as [10] and [11], our work shows more efficiency in dealing with the workspace that is not fully filled with spill. Specifically, we perform an image segmentation strategy before planning the path for coverage operation, thus the USV only travels within the area of spill. Meanwhile, the segmentation method shows strong capability in handling a coarse boundary case that has long been the challenge in the research community.

A. Image Segmentation

We first process the image by applying a median filter and BGR2GRAY methods [12] and convert the color image into a gray-scale version, as indicated by Step 1 in Fig. 4. Based on the gray-scale image, we then use the Canny operator to extract the boundary of the spill [13]. For some spill with coarse and obscure boundary, we propose to apply polynomial fitting and approximate the boundary with a spline. These are summarized as Step 2. In Step 3, since the spline plus image border forms a closed shape, we then differentiate the target area, which has a much of the spill, from the background by referring to the average gray value. Usually, the background such as water has much higher greyscale (intensity) than the

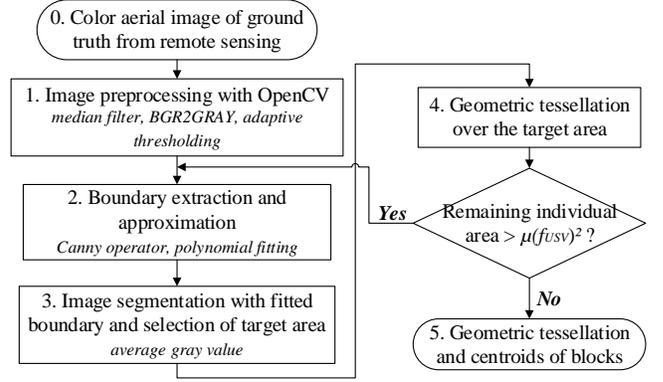


Fig. 4. The flowchart of image segmentation and workspace tessellation, along with the techniques used.

spill, such as oil and algae. Eventually, we perform a geometric tessellation to the target area with squares in Step 4. Since there may be small patches outside the target area or are not covered by tessellation blocks, we examine the remaining area of the original image after subtracting the tessellated area. If some large spill patches remain, we can perform boundary extraction again until no large spill patch can be detected in the image. The coefficient μ refers to the ratio of the area of one remaining spill patch to the area of a tessellation block $(h_{USV})^2$. Ultimately, the tessellation outcome is produced in Step 5.

Noticeably, from Step 2 to Step 5 we confirm that our strategy can deal with spills with a coarse boundary. We first approximate the spill boundary with image processing. Thus we remove a large portion of spill. By adjusting the coefficient μ and re-doing Step 4, we are able to cover the remaining spill patches and achieve the coverage with high completeness.

We evaluate the goodness of boundary approximation with both of the coefficient of determination R^2 and the Root mean square error (RMSE), defined as below. In our test, the R^2 and RMSE values are indicated in every specific figure in the third column of Fig. 5.

$$R^2 \triangleq 1 - \frac{SS_{res}}{SS_{tot}}, \quad RMSE \triangleq \sqrt{\frac{\sum_{i=1}^T (y_i - f_i)^2}{T}}, \quad (6)$$

where $SS_{res} = \sum_{i=1}^T (y_i - f_i)^2$, $SS_{tot} = \sum_{i=1}^T (y_i - \bar{y})^2$. Here, y_i refers to the y coordinate of a point on the extracted spill boundary, f_i refers to the y_i values after fitting, \bar{y} means the mean of y_i , and T refers to the total number of pixels along x -axis.

B. Geometric Tessellation

We perform geometric tessellation with squares to the areas of interest, as square can realize a high coverage rate than circles [14]. The squares are of uniform dimension and the length of side equals h_{USV} , which is proposed in Sec. III. The tessellated areas can be covered by the USV if it visits all the centroids of the tessellation blocks, namely waypoints. Therefore, the spill coverage problem is formulated as a TSP. The shortest path for such travel can be obtained by solving

this TSP and an efficient solver for the problem is elaborated in Sec. V.

V. PATH PLANNING OF THE USV

A. Multi-Goal Path Planning Problem and SOM Approach

In this section, we discuss the generation of an efficient path for USV visiting all the tessellation blocks and performing spill removal operation. Since we want the USV to return to the original location after the operation, for the purpose of replacing the filters and recharging the robot, a closed path is desired. Provided that the centroids of tessellation blocks are known, it becomes a multi-goal path planning problem. The concept of multi-goal path planning is defined in terms of finding a shortest and closed path for a given set of goals in the workspace. This problem is inspired from planning of robotic manipulators where multiple goals have to be attained by the robot in an effective time period [15].

The problem of multi-goal path planning in spill cleaning USV can be conceived as a TSP where each tessellation square represents a domain or goal point. As the TSP is modeled as uni-directed and weighted graph leading to higher computational complexity, a property of self-organizing map (SOM) method is utilized to obtain a near optimal solution to it.

SOM is an artificial neural network based solution that can realize dimensionality reduction using competitive learning [16]. The update function for neuron v with the weight vector of Euclidean distance W_v in SOM is shown below:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s)), \quad (7)$$

where s refers to the step index, t refers to training sample index, u represents is the index of the best matching unit for the input vector $D(t)$, $\alpha(s)$ is the learning coefficient, and $\theta(u, v, s)$ is the neighborhood function reflecting the distance between neurons u and v in the step indexed by s .

The SOM approach is widely used in meteorology and oceanography [17] and other geological analyses. In this work, the SOM approach is used in planning the path for spill coverage, which results are shown in Sec. VI. The current study makes a first attempt towards understanding the effectiveness of the proposed SOM path planner for the concept of spill cleaning.

B. Benchmarking

In order to benchmark the proposed SOM-based path planner, it is compared against the greedy algorithm [18] and 2-Opt algorithm [19]. Both the greedy algorithm and the 2-Opt algorithm are well-studied and widely applied in TSP related cases. 2-Opt is one of the most basic and diffusely used local search heuristics for the TSP. Additionally, 2-Opt shows its significance, particularly by achieving good results of TSP in terms of running time and length of the path.

VI. RESULTS AND DISCUSSIONS

In this section, we validate our proposed solution with five real images from Google Images, ranging from oil spill in the ocean to algae patches in the lake. The five images are numbered and shown in the first column of Fig. 5. For the rest of the columns, we demonstrate the image segmentation process and tessellation results described in Fig. 4. The images in the first column of Fig. 5 are the original color image containing ground truth, which is referred by *Step 0* in Fig. 4. The second column shows the results of boundary extraction with the Canny operator after converting the color image into a grey-scale one (*Step 2* in Fig. 4). The third column demonstrates the approximated boundary of the area of interest (*Step 3* in Fig. 4). From *image2* and *image5*, we can see that the approximated boundary fairly delineated the area of the spill in the presence of a coarse boundary. The last column shows the outcome after square tessellation with the gray-scale image as background (*Step 4* in Fig. 4). From the tessellation outcome of *image5*, we can see some large spills outside the approximated boundary are covered with a few extra tessellation blocks. It is because we set $\mu = 10\%$ and have to run *Step 2-4* multiple times to get the tessellation outcome from *Step 5*. All the tessellation blocks are of the same size in Fig. 5, which is $h_{USV} = 40$ pixels. Since the dimension of the tessellation block depends on the total pixels of the image, and considering that the original images listed are of different sizes in pixels, the tessellation blocks in the last column of Fig. 5 look different in presentation. The size of the tessellation block reflects the physical dimension with a uniform scale.

We execute the image processing as well as path planning algorithms on NVIDIA Jetson Nano™, which is a single board computer with applicable size, weight, and power consumption for the control of small-size robot such as our USV. The NVIDIA Jetson Nano is equipped with a Quad-core ARM Cortex-A57 processor, 4 GB 64-bit LPDDR4 RAM, and 16 GB eMMC Flash.

The generated paths for each image in Fig. 5 are shown in Fig. 6 using SOM approach with $h_{USV} = 40$ pixels. The computation cost and other facts of path planning after running on Jetson Nano are summarized in Table I. For an image with more pixels, it deserves more tessellation blocks as the image represents a larger physical area, and hence higher computation cost. For instance, it took 105.97 seconds for *image5* to generate a sub-optimal path and meanwhile, more neurons and iteration times are needed. However, *image3* took 105.97 seconds to get the path while using only 8% number of neurons of *image5*. We did not specify the starting point of the USV in Fig. 6, as the USV may start from any tessellation block. Even if the USV is outside the spill, its traveling distance to the closest block is negligible comparing to the overall path length.

A wider operation range h_{USV} helps reduce the number of tessellation blocks as well as computation costs. The outcomes of having $h_{USV} = \{30, 40, \text{ and } 50\}$ are shown in Table II,

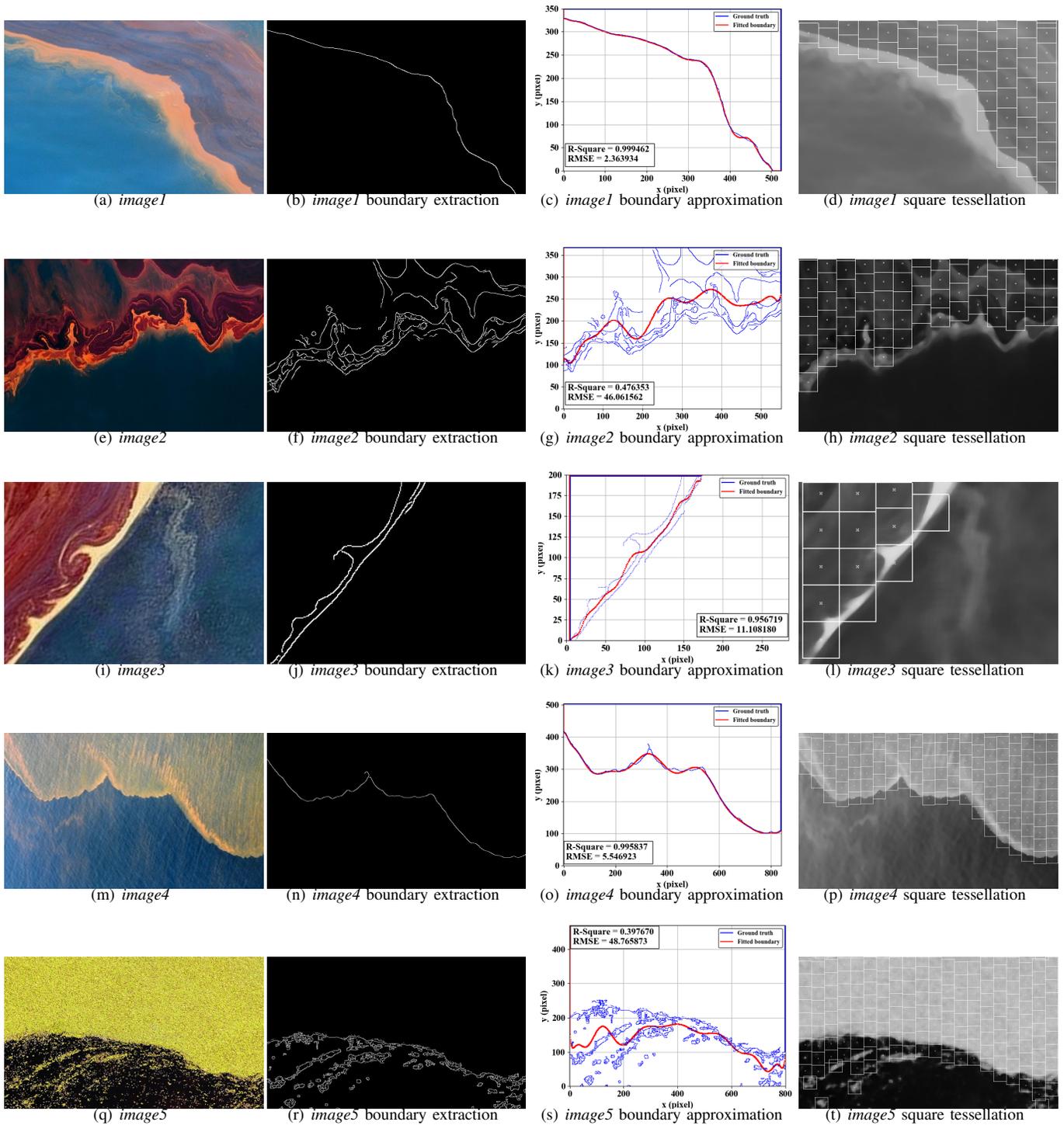


Fig. 5. The procedure of the image segmentation using five different real aerial images. The real images are presented in the first column. The outcomes after boundary extraction are shown in the second column, while the approximated boundary is shown in the third. The last column demonstrates the results after applying square tessellation with $h_{USV} = 40$ pixels.

TABLE I
COMPUTATION COSTS AND OUTCOMES OF PATH PLANNING WITH FIGURES FROM FIG. 5 USING SOM APPROACH ($h_{USV} = 40$ PIXELS).

Metrics	<i>image1</i>	<i>image2</i>	<i>image3</i>	<i>image4</i>	<i>image5</i>
<i>Computation time (s)</i>	76.18	83.36	62.51	98.38	105.97
<i>Neurons created #</i>	392	504	104	1112	1416
<i>Iterations #</i>	19901	20738	15478	23376	24181
<i>Total length of path (pixels)</i>	2080.779	2643.399	560.871	5824.980	7330.534

TABLE II
COMPUTATION COSTS AND OUTCOMES OF PATH PLANNING WITH *image1* AND *image3* FROM FIG. 5 WITH $h_{USV} = \{30, 40, \text{AND } 50\}$ PIXELS, BY USING SOM, GREEDY ALGORITHM, AND 2-OPT ALGORITHM.

Images	h_{USV} value	SOM				Greedy		2-Opt	
		C' time (s)	N' s#	Itr' s #	P' length	C' time (s)	P' length	C' time (s)	P' length
<i>image1</i>	30	89.09	672	21697	2666.976	2.39	3147.935	90.05	2660.207
	40	76.18	392	19901	2080.779	0.46	2502.363	18.78	2102.835
	50	72.98	248	18375	1681.480	0.11	2030.552	4.44	1667.099
<i>image3</i>	30	66.83	160	16914	654.213	0.03	618.911	1.09	655.774
	40	62.51	104	15478	560.871	0.01	506.634	0.3	560.472
	50	58.26	72	14235	480.786	0.01	417.799	0.09	480.786

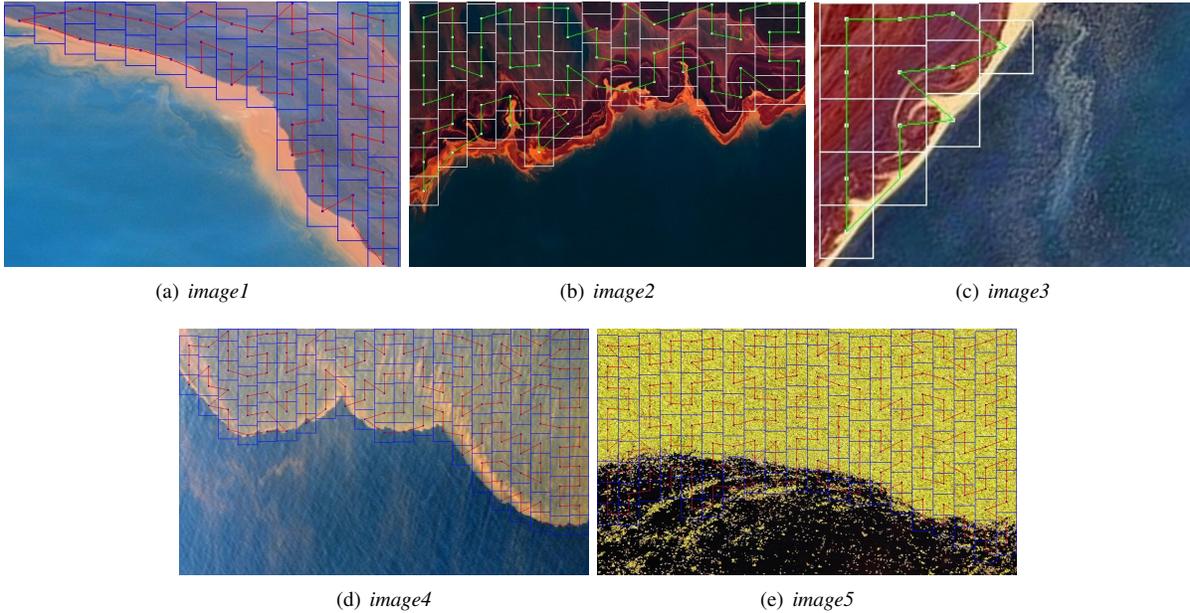


Fig. 6. Planned paths for *image1*–*image5* using SOM with $h_{USV} = 40$ pixels.

after testing with *image1* and *image3*. Beside less computation time, denoted as C' time in Table II, a greater h_{USV} value also leads to a shorter path length, denoted as P' length in Table II. The planned paths of USV by having different h_{USV} values are depicted in Fig. 7.

To evaluate the performance of SOM in generating the sub-optimal path, we also present the results by applying the greedy algorithm and 2-Opt algorithm. We still test with *image1* and *image3*. The quantitative results are presented in the rightmost part of Table II, while the planned paths are presented in Fig. 8. By comparing Fig. 8 with Fig. 7(a) and Fig. 7(c), we can see that SOM approach generates a smoother path than the other two, although it takes much longer

computation time. The greedy algorithm typically yields the longest path length, yet the shortest computation time. But for the cases with very few tessellation blocks, the Greedy algorithm may yield a better solution, as indicated by *image3* in Table II. 2-Opt algorithm maintains a good balance between computation time and total path length, but it becomes slower than SOM if more tessellation squares are involved. We would suggest using the greedy algorithm for computation time sensitive tasks, and 2-Opt algorithm for situations with less tessellation blocks or waypoints. SOM approach can usually yield a better solution than many other solvers, and this further strengths its wide application in oceanography and geographic analysis.

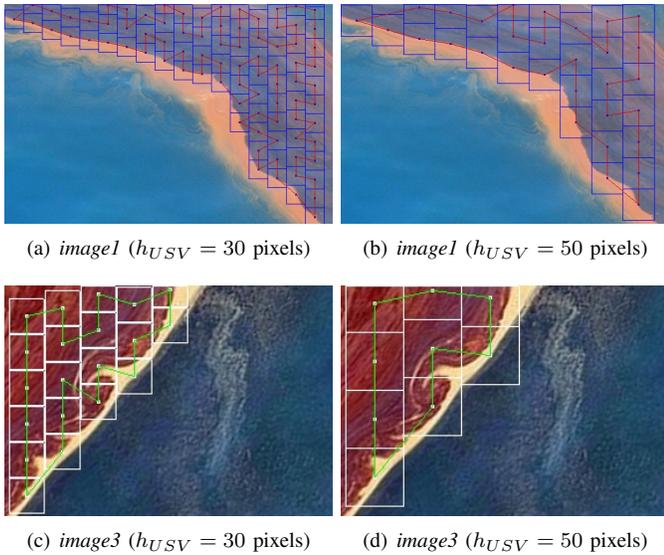


Fig. 7. Planned paths for *image1* and *image3* using SOM but with different h_{USV} values.

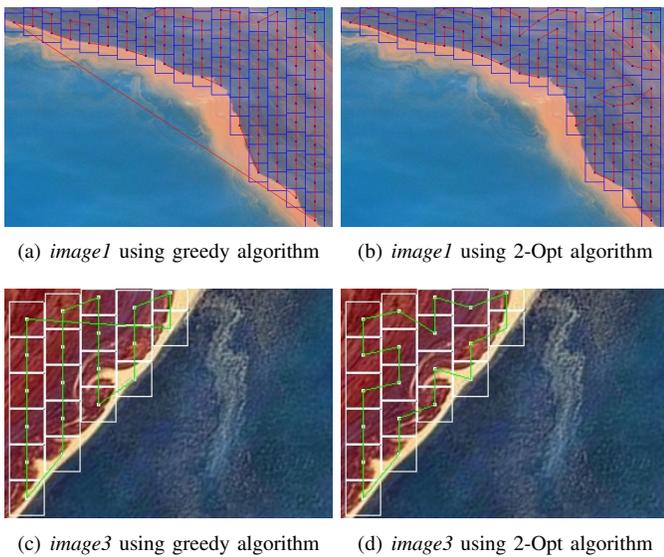


Fig. 8. Planned paths for *image1* and *image3* using greedy algorithm and 2-Opt algorithm, with a uniform $h_{USV} = 30$ pixels.

VII. CONCLUSION AND FUTURE WORKS

This paper presents an image processing and spill cleaning model based efficient path planning method using SOM approach. Specifically, the image processing is used to produce a more targeted tessellation to the workspace, while the spill cleaning model derived from a USV collecting spill in maneuvering is used to determine the dimension of the tessellation blocks. The planned path is obtained by applying SOM approached, and its performance is evaluated by comparing with the greedy algorithm and 2-Opt algorithm. This solution is proved to be capable in dealing with a diversity of oil spills, especially the spills with coarse boundary, featuring a deterministic tessellation block size h_{USV} .

In our future work, we will mainly focus on USV implementation and field tests in order to validate our solution systematically. In addition, for a large-scale spill, we will explore a multi-USV system based solution and deploy multiple USVs in operation. Meanwhile, we will incorporate an unmanned aerial vehicle (UAV) into the existing solution and realize a fully autonomous spill removal system.

VIII. ACKNOWLEDGMENT

Yogang Singh was supported in part by the Arequipa Nexus Institute.

REFERENCES

- [1] E. Pinto, P. Santana, F. Marques, R. Mendonça, A. Lourenço, and J. Barata, "On the design of a robotic system composed of an unmanned surface vehicle and a piggybacked vtol," in *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2014, pp. 193–200.
- [2] J. Almeida, A. Ferreira, B. Matias, A. Dias, A. Martins, F. Silva, J. Oliveira, P. Sousa, M. Moreira, T. Miranda *et al.*, "Air and underwater survey of water enclosed spaces for vamos! project," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–5.
- [3] S. Bella, A. Belbachir, and G. Belalem, "A hybrid architecture for cooperative uav and usv swarm vehicles," in *International Conference on Machine Learning for Networking*. Springer, 2018, pp. 341–363.
- [4] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.
- [5] J. H. Bae, S. Luo, S. S. Kannan, Y. Singh, B. Lee, R. M. Voyles, M. Postigo-Malaga, E. G. Zenteno, L. P. Aguilar, and B.-C. Min, "Development of an unmanned surface vehicle for remote sediment sampling with a *van veen* grab sampler," in *2019 MTS/IEEE OCEANS*. IEEE, 2019.
- [6] N. M. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *Journal of hazardous materials*, vol. 154, no. 1-3, pp. 880–887, 2008.
- [7] S. Jung, H. Cho, D. Kim, K. Kim, J.-I. Han, and H. Myung, "Development of algal bloom removal system using unmanned aerial vehicle and surface vehicle," *IEEE Access*, vol. 5, pp. 22 166–22 176, 2017.
- [8] W. Jo, J. H. Park, Y. Hoashi, and B.-C. Min, "Development of an unmanned surface vehicle for harmful algae removal," in *2019 MTS/IEEE OCEANS*. IEEE, 2019.
- [9] W. Jo, Y. Hoashi, L. L. P. Aguilar, M. Postigo-Malaga, J. M. Garcia-Bravo, and B.-C. Min, "A low-cost and small usv platform for water quality monitoring," *HardwareX*, vol. 6, p. e00076, 2019.
- [10] X. Jin and A. Ray, "Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments," *International Journal of Control*, vol. 87, no. 4, pp. 787–801, 2014.
- [11] M. Penmetcha, S. Luo, S. Arabinda, J. E. Dietz, B. Yang, and B.-C. Min, "Computer vision-based algae removal planner for multi-robot teams," in *2019 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2019.
- [12] S. Kapur and N. Thakkar, *Mastering OpenCV Android Application Programming*. Packt Publishing Ltd, 2015.
- [13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [14] S. Luo, J. H. Bae, and B.-C. Min, "Pivot-based collective coverage control with a multi-robot team," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 2367–2372.
- [15] T. Danner and L. E. Kavasaki, "Randomized planning for short inspection paths," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 971–976.
- [16] T. Villmann and H.-U. Bauer, "Applications of the growing self-organizing map," *Neurocomputing*, vol. 21, no. 1-3, pp. 91–100, 1998.
- [17] Y. Liu and R. H. Weisberg, "A review of self-organizing map applications in meteorology and oceanography," 2011.

- [18] W. Bednorz, "Advances in greedy algorithms," *Vienna: I-Tech Education and Publishing KG*, vol. 14, no. 6, 2008.
- [19] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-opt algorithm for the tsp," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1295–1304.