

Article

Two-Dimensional Path Planning Platform for Autonomous Walk behind Hand Tractor

Padma Nyoman Crisnapati  and Dechrit Maneetham *Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi,
Pathum Thani 12110, Thailand

* Correspondence: dechrit_m@rmutt.ac.th

Abstract: The use of autonomous vehicles in agriculture has increased in recent years. To fully automate agricultural missions, particularly the tillage process using the walk-behind hand tractor, the path planning problem for the robot must be solved so that all points in the intended region of interest may be traced. The current planning algorithm has been successful in determining the best tillage path. On the other hand, the algorithm ignores the path built using the dynamic starting point, finish point and path distance. We propose a path planning concept for back-and-forth path patterns. Our algorithm employs a novel approach based on Laravel and Google Maps, which considers the user's desired distance interval, start point, and finish point. We demonstrated auto-generating vertex-edge pathways in this research. Field trials using a walk-behind hand tractor in a plowing mission have been successfully conducted to validate the accuracy of the resulting waypoint coordinates.

Keywords: path planning; back-forth path; hand tractor; tillage; unmanned ground vehicle



Citation: Crisnapati, P.N.; Maneetham, D. Two-Dimensional Path Planning Platform for Autonomous Walk behind Hand Tractor. *Agriculture* **2022**, *12*, 2051. <https://doi.org/10.3390/agriculture12122051>

Academic Editors: José Lima, José Boaventura Ribeiro da Cunha, Antonio Valente and Filipe Neves Dos Santos

Received: 17 October 2022
Accepted: 23 November 2022
Published: 29 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mobile robots have been effectively used to perform vital unmanned operations in various situations during the last few decades, including military, industrial, security, and agricultural operations [1]. Mobile robots are increasingly being used in modern agriculture because the number of farmers is decreasing, necessitating more effective farming practices through agricultural mechanization. Agriculture has been mechanized in most field operations, including tillage, transplantation, agrochemical application, harvesting, and drying. While most agriculture operations are being explored for automation, Indonesian farmers' tillage methods are not yet automated.

Path Planning [2] is a critical problem that must be solved before a mobile robot may navigate and explore autonomously; this kind of robot is usually known as an Unmanned Ground Vehicle (UGV). The UGV can search for pathways based on their start and finish points, the surrounding environment, and specified parameters. With path planning, the UGV can save time and significantly reduce the UGV's wear and tear and the associated costs. The critical significance of path planning for UGVs is an intriguing subject of study.

UGV applications have exploded in popularity during the last five years. Survey missions for two-dimensional (2D) coverage have demonstrated exceptional performance among diverse uses. For instance, two-dimensional and three-dimensional mapping [3], search and rescue [4] in combination with an Unmanned Aerial Vehicle for disaster and emergency management [5], or precision agriculture [6]. The survey mission performed by UGVs in agriculture is generally separated into two stages: (i) the preparation stage, during which the vehicle is selected, the embedded system configured, and the path is planned; and (ii) the execution stage, during which the vehicle operates autonomously and gathers data. Path planning must be completed to fully automate the operation, which is described as calculating paths for the robot to traverse the Region of Interest (ROI) [7]. The problem's complexity has been determined as NP-Hard [8]. However, assuming that traveling in

a straight line is the quickest and most efficient way to traverse the entire landscape, a sophisticated solution involves reducing the problem and computing the path with the least tracing, e.g., [9,10]. While several of these studies have produced significant results, they do not account for the path factors associated with the user's start and finish points. The puddler widths can be changed because the UGV is configured as a walk-behind hand tractor. The path planning must account for additional criteria such as interval tillage line distance from the user. Taking these three factors into account will affect the accuracy of land management while utilizing various models of tractors; this requires a path design that considers the mission's distance interval and the mission's starting and ending sites.

Due to limited field resources, efficient path computation is required; this can be accomplished by moving the tractor in a straight line, forming a back-and-forth path (BFP) [11]. This study analyses convex polygon boustrophedon routes, a fast algorithm for determining BFP coverage on an ROI, considering the distance interval, start, and finish points (Figure 1). The decomposition of boustrophedon cells is a widely used technique for coverage. The cells of the boustrophedon are filled in a simple backward and forward motion. Once each cell is closed, the entire environment is sealed [12]. As a result, the scope is narrowed to identify the whole path through the graph, reflecting the boustrophedon decomposition's cell proximity connection. This strategy is well-suited for plowing missions with defined start and finish points and coverage.

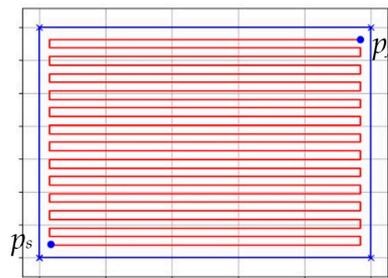


Figure 1. Proposed concept for path planning, the red line represents the generated path based on the starting point (p_s), finish point (p_f), and the ROI (blue polygon).

Over the last few decades, path planning problems have been addressed extensively; nevertheless, the available literature focuses primarily on UGV and UAV difficulties. Various strategies and algorithms can be employed with ground robots; generally, path planning based on ROI can be classified as solid or border representations. NP-Hard problems are used as a base to path planning with solid-representation approaches; emphasis is on interiors rather than regions [8]. Generally, the interior of an environment is represented as a grid of binary values or probabilities [13]. This method is also known as the grid-based method [7]. This resolution-complete approach takes a significant amount of computer power for high-resolution maps. The grid-based process makes use of a neural network in which the region is the input [14], divides it into triangular cells, and in conjunction with a region growth algorithm, generates a path [3], utilizing a Turing machine [15] or by using a genetic algorithm [16]. In addition to solid representation, the boundary representation approach defines the region of interest (ROI) depending on the geometry of the polygon. The polygonal model is adopted in this study. Numerous literature evaluations exist for alternative path-planning approaches [7].

One of the most extensively used polygonal representations is boustrophedon cellular decomposition (BCD); this method provides a path-planning solution for traversing the polynomial world [12]. BCD generates a line using the idea of cells and then explores each cell using this manner, except that the robot's direction of movement is always the same. Huang [9] developed another coverage approach for mine-operating robots that uses alternate orientations perpendicular to the ROI range instead of polygons. However, this method ignores the trip distance between the starting point and the ROI. Numerous studies have been conducted on the covering of non-convex terrain. This strategy's fundamental

difficulty is acquiring a near-optimal ROI partition and mapping coverage path to visit each division sequentially [17]; in some circumstances, other restrictions such as energy consumption are incorporated into the optimization process [18]. In this paper, we focus exclusively on the convex region.

Path planning with UGVs has been generally approached from the standpoint of land-based mobile robotics, but with novel functionalities. Most approaches have adhered to Huang's optimality criteria [9], where the ideal path is the one with the fewest paths. The techniques for calculating the convex polygons area are provided by [19]. The convex covering area method is presented in research [10]. Numerous researchers have also used path planning in agriculture, including UGV, to determine the ground features of greenhouses [20] by utilizing the BFP technique and a differential robot.

Further, [21] created a robot for precision pollination in a greenhouse and implemented it on a differential robot. Path planning for an unmanned ground vehicle in conjunction with aerial images utilizing the A* search method in graphs with gradient optimization of the descent to smooth the trajectory [22]. Large tractor machines are used; this work optimizes the harvest area of a combined harvester robot for wheat or rice using convex and concave polygon fields [23].

In short, the prior approach ignored the dynamic size of the tractor puddler, limiting its application to a single vehicle, and the approach was limited to establishing the beginning point and the tractor with huge dimensions. As a result, the initial approximation is inadequate. A more accurate path can be obtained by combining the dynamic starting inputs for the interval distance, start point, and finish point and then applying these to the walk-behind tractor. This paper is a continuation of our previous work, TROLLS: Tractor Controlling System for Walk-Behind Hand Tractors, presented in its first version [24,25]. The initial version can control the tractor remotely without automation. This article presents novel results in an open-source web-based platform and Google Maps for planning rectangular polygon paths that consider the width of the puddler in addition to the mission distance interval's start point and end point as criteria. This path planning is the first step towards an autonomously operating tractor engine validated in field trials using the tractor G1000 manufactured by Quick. The plowing mission was carried out as a validation of the success of the path-planning platform. The tractor used in the field test is equipped with an embedded system platform that allows the tractor to move autonomously based on the path planning platform's waypoints.

2. Materials and Methods

2.1. General Definition and Notation

The tractor in this study moves in a two-dimensional planar plane, where L is a linear combination (Figure 1), as in the Formula (1). The distance between point a and line L is the perpendicular distance between c and a point on L (2). The distance between parallel lines L_1 and L_2 is denoted by (3). Equation (4) shows a line segment as a horizontal line linking two points, c and d [26]. Abbreviations and symbols used in the article are listed in Table 1.

$$[c, d] = L(\alpha) = (1 - \alpha) c + d \rightarrow \alpha \in R \quad (1)$$

$$dst(a, L) = dst(a, b) \rightarrow b \in L, [c, d] \perp L \quad (2)$$

$$dst(c \in L_1, L_2) \quad (3)$$

$$\overline{cd} = T(\alpha) = (1 - \alpha) c + \alpha d \rightarrow 0 \leq \alpha \leq 1 \quad (4)$$

2.2. Region of Interest (ROI)

As seen in Figure 1, The ROI in this study is in the form of a 2D planar (convex polygon) (5), where a collection of vertices and edges is represented by the Formulas (6) and (7) [11,26]. L is the support line formed by the intersection with the polygon boundary

line and forms a pair of antipodal points. The distance between the two support lines is created on the same polygon and is named width [19].

$$Q = \{V, E\} \quad (5)$$

$$V = \{1, \dots, n\} \quad (6)$$

$$E = \{(1, 2), \dots, (n - 1, n), (n, 1)\} \quad (7)$$

Table 1. Abbreviations and symbols are used in this article.

Symbol	Description	Symbol	Description
R^2	Two-dimensional (2D) planar tractor work area	φ_1	Latitude of the initial point
(x, y)	Within the work area, a point is a location	φ_2	Latitude of the final point
L	Line as a linear combination of two points	$\Delta\varphi$	$\varphi_2 - \varphi_1$
a, b, c, d	Four points within a Region of Interest (ROI)	λ	Longitude
Q	The Region of Interest	λ_1	Longitude of Initial Point
V	A collection of points on a plane (vertices)	λ_2	Longitude of Final Point
E	A group of edges	$\Delta\lambda$	$\lambda_2 - \lambda_1$
$A(Q)$	Polygon area	<i>earth's_radius</i>	6371 km
I_x	Tillage footprint length	d	Distance between two points
I_y	Tillage footprint width	ps	Starting point
p	Tractor position	pf	Finish point
$p1, p2$	Two points on the coverage area of the puddler	$p0, \dots, pn$	Knot edge path
$C(T)$	The coverage area of a tillage line	dx	Distance between tillage line
φ	Latitude	δ	Declination angle
dst	Distance between two parallel lines		

2.3. Unmanned Ground Vehicle

When the tractor is used for tillage operations, it is equipped with a puddler that points directly to the ground at a consistent harrow height. The tractor location is defined as $p = (x, y)$ in R^2 , which allows for the definition of the tractor path as $s(t): R \rightarrow R^2$.

2.4. Two-Dimensional Leveller/Puddler Footprint

When the tractor plows an area, the tractor departs from point ps , following the path by carrying a puddler and heading to the finish point pf (see Figure 1). During the tillage process, the puddler forms an area known as the tillage footprint of $I_x \times I_y$ so that the tillage footprint can be represented by Equation (8). If s is the tillage path, then Equation (9) is the coverage area (ρ) of the puddler, and each waypoint of Q is represented in Equation (10). The tractor tillage mission overview is presented in Figure 2.

$$Tl(p) = I_x \times I_y \quad (8)$$

$$C(s) = \cup p \in s Tl(p) \quad (9)$$

$$A(Q) \subseteq C(\rho) \quad (10)$$

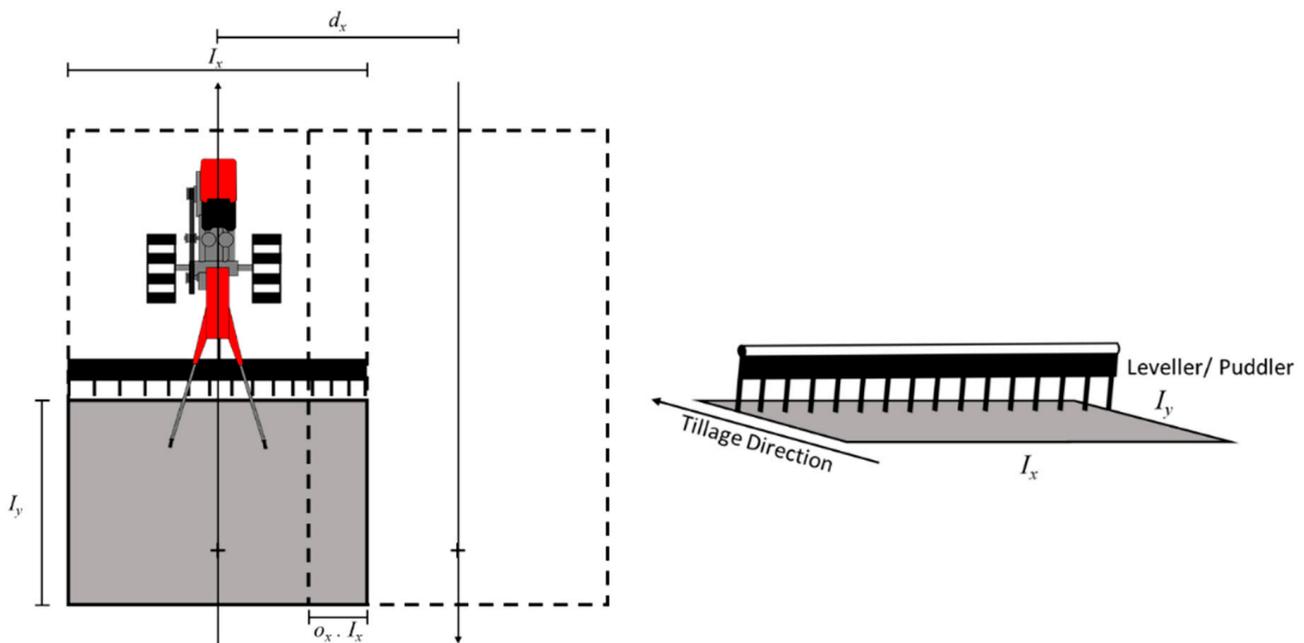


Figure 2. Walk-behind hand tractor tillage mission from a perspective view. The top perspective is depicted on the left. Lines with arrows indicate the direction of tillage. The gray box denotes the area crossed by the puddler, referred to as the $Tl(p)$ tillage footprint. After completing one tillage line, the tractor will proceed to the next. d_x denotes the distance between tillage lines. The right image depicts a view from a lateral position.

2.5. Back and Forth Path (BFP)

In path planning, many possible path forms are created [8]. Therefore, particular patterns such as spirals, zigzags, stars, or back and forth are the solution to these problems. In this study, the BFP is implemented on a tractor that moves in a straight line at the ROI. The advantages of BFP allow the tractor to keep the puddler stable and make it easy for autonomous vehicles to follow the tillage line.

Tillage line T is a linear combination Equation (11), which means that $\alpha \in R$ and $0 \leq \alpha \leq 1$ as seen in Figure 3. $C(T)$ is the region enclosed by T_{left} and T_{right} . These two lines are parallel to T by an $\frac{I_x}{2}$ distance. When tillage occurs at an ROI, one tillage line is insufficient to cover the entire area, so plowing with a BFP pattern is used in Equation (12). Equation (13) is used to calculate the coverage area of Q .

$$T(\alpha) = (1 - \alpha) p_1 + \alpha p_2 \tag{11}$$

$$P = \{T_1, \dots, T_n\} \tag{12}$$

$$A(Q) \subseteq \cup F \in P C(T) \tag{13}$$

These calculations deduce that BFP is a collection of sequential points used as a plowing route boundary or waypoint, which we connect with a straight line between tillage lines. This research aims to develop a path planning platform for rectangular ROI convex polygons and validate the autonomous walk-behind hand tractor’s algorithm.

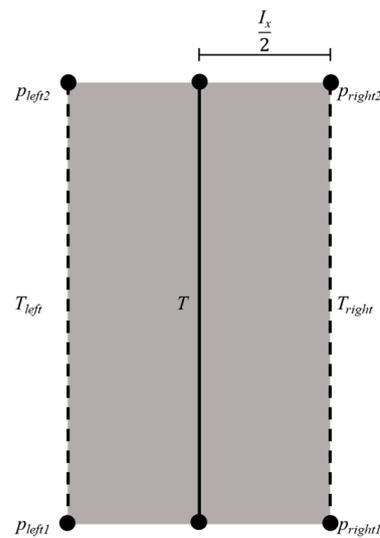


Figure 3. Area covered when the tractor crosses the T tillage line.

3. Embedded System Platform

Walk-behind tractors (affectionately referred to as walking tractors, pedestrian-controlled tractors, or power tillers) are prevalent in small and medium-sized rural communities. This type of tractor offers several advantages, including its low price and ease of use. Additionally, these tractors can perform various agricultural tasks, including harvesting, crop protection, irrigation, threshing, and transporting [27]. A G1000 Quick tractor equipped with a boxed embedded control system was used in this study. The tractor’s wheels turn due to a clutch handle being pulled, as illustrated in Figure 4. This platform is used to validate the path planning platform.

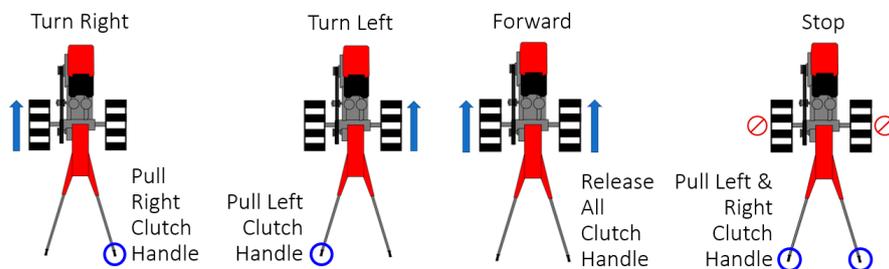


Figure 4. G1000-Quick Walk-Behind Tractor movement control.

An autonomous navigation system enables tractors to navigate between waypoints autonomously [28]. We are attempting to construct an additional validation platform. First, the list of waypoints generated by the platform informs of the final Global Positioning System (GPS) coordinates (longitude and latitude). The platform will then build a set of reference pathways (straight lines) that connect each waypoint. Additionally, these coordinates are used in a control system to direct the robot’s movement between waypoints along the reference route. The results obtained from the GPS and IMU sensors contain significant noise. This will interfere with the tractor’s movement system; hence, the application of a filter in this system is essential. As an early observation, the Kalman Filter and the Butterworth Filter were compared. The results of applying these two techniques to both sensors are depicted in Figure 5. It is evident from the image that the application of the filter effectively reduces the noise and error that arises. Kalman filter was selected because of its superior input stability for the controller. The controller, not the tractor, determines the distance and heading. Figure 6 depicts the relationship between GPS sensors, IMU (Magnetometer), and Waypoint Techniques as inputs and outputs rather than controllers such as servo motor drives.

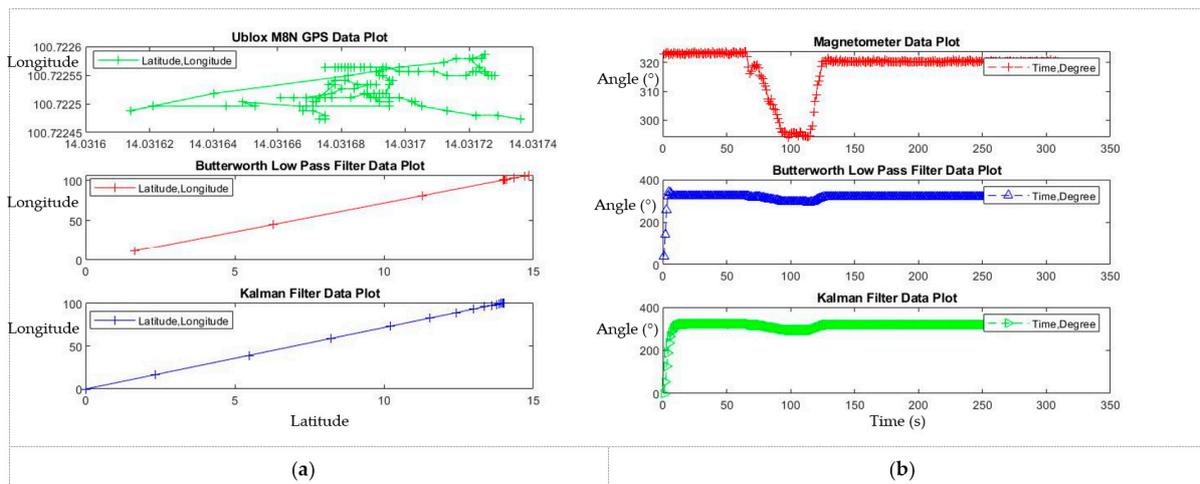


Figure 5. Comparison of noise and error correction data plot using filter. (a) GPS Data Plot; (b) Magnetometer Data Plot.

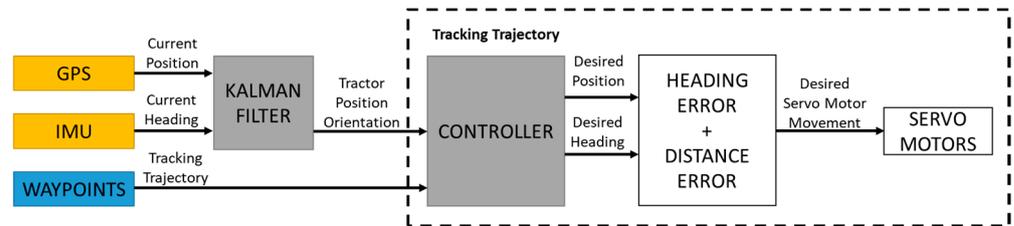


Figure 6. Waypoint system overview.

GPS technology is used to determine the current location of the tractor using a GPS receiver [29]. GPS technology is the backbone of the autonomous vehicle because it receives GPS coordinates from the location; the U-Blox Neo-M8N type sensor is used in this study. As indicated in Table 2, this type of GPS was chosen because it is economical for Indonesian farmers and offers a reasonable degree of precision.

Table 2. Neo M8N first fix and horizontal accuracy.

Parameter	Condition	GPS	GLONASS	BeiDou
First fix time	Cold start	29 s	26 s	27 s
	Hot start	1 s	1 s	1 s
Horizontal accuracy	Autonomous	2.5 m	2.5 m	2.5 m
	SBAS	2.0 m	2.0 m	2.0 m

After the tractor location is obtained, the distance between the current tractor position and the target position is calculated using the Haversine Equations (14)–(16) [30]. In addition to distance, bearing angle is an essential metric for autonomous robotics as it indicates the robot’s direction and helps the tractor follow the correct path [30]. The angle of the tractor bearing concerning the initial latitude and longitude coordinates can be calculated using the Equations (17)–(19).

$$x = \sin^2(\Delta\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2 (\Delta\lambda/2) \tag{14}$$

$$y = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1 - a}) \tag{15}$$

$$d = y \cdot \text{earth's_radius} \tag{16}$$

$$x = \sin(\Delta\lambda) \cos(\phi_1) \tag{17}$$

$$y = \cos(\varphi_1) \cdot \sin(\Delta\varphi_2) - \sin(\varphi_1) \cdot \cos(\varphi_2) \cos(\Delta\lambda) \tag{18}$$

$$\text{Bearing Angle} = \text{atan2}(x, y) \cdot 180 / \pi \tag{19}$$

The heading angle is obtained from the magnetometer sensor found on the IMU. This sensor can provide an absolute correct direction, but its efficiency is reduced when a magnetic field other than the Earth’s magnetic field is present [31]. One source of significant magnetometer error is the annual variation in the tilt of the Earth’s axis of rotation and the Earth’s rotation around the sun, also known as the declination angle. The declination angle can be determined using the Equation (20), where a day is the number of days remaining before January. The declination angle should be added to the sensor-derived compass direction [31]. This inaccuracy varies according to the sensor’s location and can be found at <https://www.magnetic-declination.com/> (1 February 2021). For instance, in Gianyar, Bali- Indonesia, the declination is 0°43’ EAST (POSITIVE). Autonomous UGVs require knowledge about their travel direction. The heading angle in the horizontal plane is defined as the clockwise angle from true north [30]. As a result, headings that rotate between 0° and 360° refer to true north (21).

$$D = \sin - 1(\sin(23,45^\circ) \cdot \sin(360/365 \cdot (\text{day}-81))) \tag{20}$$

$$\text{heading} = \text{atan}(Yh / Xh) \tag{21}$$

4. Results and Discussion

This research aims to compute the coverage path Equation (22) to obtain an edge-vertex path with a boustrophedon cellular decomposition pattern and test it on the Quick G1000 walk-behind tractor. Briefly, our goal is to determine the edge-vertex routes within the polygon, the start-finish points, and the distance between tillage lines.

$$\tau = \{p_s, p_0, \dots, p_n, p_f\} \tag{22}$$

Figure 7 shows a path that forms an alternate direction when the tillage line generated is parallel to one side of the ROI. An edge-vertex path (EVP) is formed inside polygon Q. The EVP is formed by the intersection of waypoints and the $L_{tillage}$ line with polygon Q. In Algorithm 1, the inputs are the polygon Q, the initial vertex b , the antipodal vertex c and d , and the distance between the tillage line d_x . d_x is the first user input determined by the size of the puddler rather than the type. At the beginning process, the $L_{tillage}$ line is parallel to the sides (a, b) that have been displaced perpendicularly toward the c and d directions. After intersecting with polygon Q, $L_{tillage}$ subtracts or adds d_x to create p_1 and p_2 . The next step is to combine the points into the path and connect them to the CalculateConnect function, which provides a perpendicular boundary between the two points. If the footprint intersects the polygon line, this method will be repeated and shifted to point c or d . This algorithm returns the path $\rho = \{p_0, \dots, p_n\}$, at the final step. EVP always begins from vertex a and sweeps towards vertex b, c , and d .

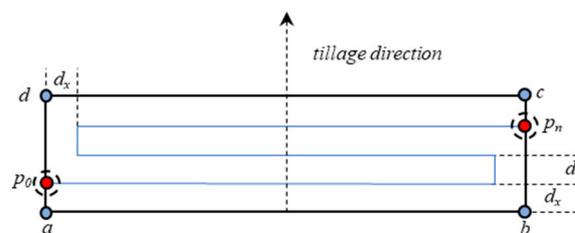


Figure 7. A diagrammatic representation of the suggested edge-vertex path algorithm for walking behind a hand tractor. The algorithm regularly adds points by recognizing the lines’ intersection with a predefined region of interest (ROI).

Algorithm 1: Polygon ROI edge-vertex identification ($\text{GetPath}(a, b, c, d)$). Calculating the BFP path begins with ($\rho = \{p_0, \dots, p_n\}$) for the polygon ($Q = V, E$). The tractor begins its movement at vertex a and sweeps in the direction of c and d .

Data: Q, dx, a, b, c, d

Result: ρ

1. $L_{tillage} \leftarrow \text{CreateLine}(a, b);$
2. $L_{tillage} \leftarrow \text{OffSet}(L_{tillage}, dx);$
3. $\rho \leftarrow \emptyset;$
4. **while** $\text{Intersects}(C(L_{tillage}); Q)$ **do**
5. **if** $(\text{IntersectEdgesLeft})$ **then**
6. $ip_1; ip_2 \leftarrow \text{IntersectEdges}(L_{tillage}; E) + dx;$
7. **else if** $(\text{IntersectEdgesRight})$ **then**
8. $ip_1; ip_2 \leftarrow \text{IntersectEdges}(L_{tillage}; E) - dx;$
9. **end if**
10. $\rho \leftarrow \text{CalculateConnect}(\rho; ip_1; ip_2);$
11. $L_{tillage} \leftarrow \text{OffSet}(L_{tillage}, dx);$
12. **end while**
13. **return** $\rho;$

The result of implementing Algorithm 1 is a Laravel-based website platform that uses Google Maps for satellite map sources. The user can add a region of interest to the accessible map and specify the interval distance in meters, and the starting and ending points. After determining all of the initial inputs, the final step is to click the Generate Paths button to generate the path automatically. A scenario with varying intervals from the same ROI, start point, and finish point is given to validate this platform. This platform is evaluated using 1, 1.5, 2, and 2.5 m interval distance values. Figure 8 shows that the platform successfully auto-generating coordinates in four different scenarios. Furthermore, the coordinates generated by the path planning platform were validated by field tests to support the autonomous walk-behind Quick G1000 tractor.

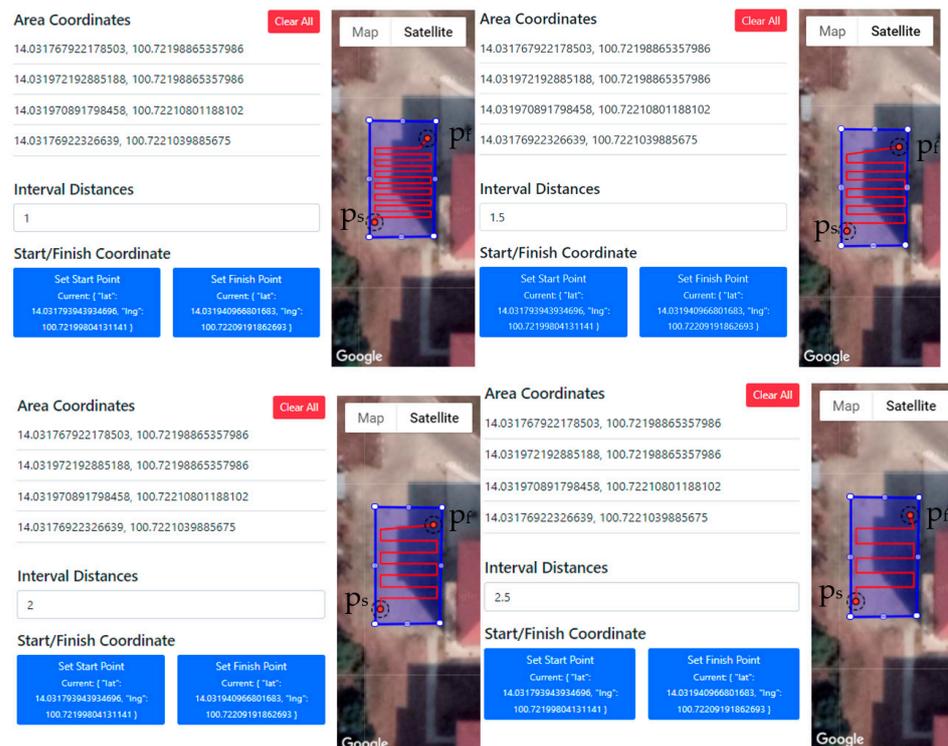


Figure 8. Path planning platform implementation using Laravel and Google Maps.

As a validation scenario, an Embedded System Platform was installed on the tractor. Figure 9 shows the Embedded System Platform box, which contains a controller and several motors that control the clutch handle and propel the tractor wheels autonomously via the waypoint method (coordinates from Path Planning Platform) at a constant and predetermined speed. In the controller, several algorithms are implanted to read heading distance, error, and angle so that the tractor can move autonomously, including Algorithms 2–4.

Algorithm 2: Compass Reading.

Data: normalize, normalize_YAxis, normalize_XAxis, heading, declinationAngle, headingDegrees

Result: headingDegrees

1. $normalize \leftarrow \text{ReadCompassNormalize}();$
 2. $heading = \text{atan2}(\text{normalize_YAxis}, \text{normalize_XAxis});$
 3. $\text{declinationAngle} = (\angle + (\text{min}/60)) / (180/\pi);$
 4. $heading - = \text{declinationAngle};$
 5. **if** (heading less than 0) **then**
 6. $heading + = 2 \times \pi;$
 7. **else if** (heading more than $2 \times \pi$) **then**
 8. $heading - = 2 \times \pi;$
 9. **end if**
 10. $\text{headingDegrees} \leftarrow \text{ConvertToDegrees}(\text{heading});$
 11. **return** headingDegrees;
-

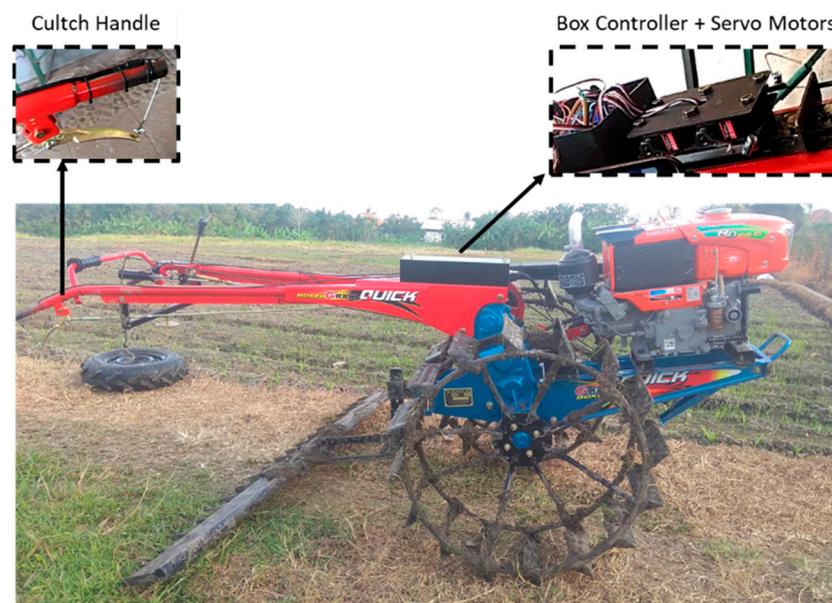


Figure 9. Quick G1000 walk-behind tractor.

Algorithm 2 is the compass sensor reading. The reading results are normalized to vector form and then used as input for the heading computation ($\text{atan2}(\text{normalize YAxis}, \text{normalize XAxis})$). To compensate for the compass's divergence from the Earth's actual north pole, the declination angle (δ), which may be found at <http://magnetic-declination.com/> (1 February 2021), must be calculated. The declination angle for Gianyar, Bali, Indonesia is $0^{\circ}43'$ EAST (POSITIVE). There are two conditions in which the heading calculation results should be corrected: larger than $2 \times \pi$ and fewer than 0 degrees. After that, the result is translated into degrees.

Algorithm 3 is a distance computation between the robot's current location and the target waypoint. The GPS sensor value represents the current location's degree of longitude and latitude, and the target waypoint must be translated to radians. Following that, the

Haversine formula is used to calculate the great-circle distance.

Algorithm 3: Calculate Distance to Target: distance from the current location to the target waypoint.

Data: deltaLongitude, currentLong, currentLat, targetLong, targetLat

Result: distanceToTarget

1. $\Delta\text{Longitude} \leftarrow \text{radians}(\text{currentLong} - \text{targetLong});$
2. $\text{lat1} \leftarrow \text{radians}(\text{currentLat});$
3. $\text{lat2} \leftarrow \text{radians}(\text{targetLat});$
4. $\Delta\text{Longitude} = \text{sq}((\cos(\text{lat1}) \times \sin(\text{lat2})) - (\sin(\text{lat1}) \times \cos(\text{lat2}) \times \cos(\Delta\text{Longitude})));$
5. $\Delta\text{Longitude} += \text{sq}(\cos(\text{lat2}) \times \cos(\Delta\text{Longitude}));$
6. $\Delta\text{Longitude} = \text{sqrt}(\text{deltaLongitude});$
7. $\text{denom} = (\sin(\text{lat1}) \times \sin(\text{lat2})) + (\cos(\text{lat1}) \times \cos(\text{lat2}) \times \cos(\Delta\text{Longitude}));$
8. $\Delta\text{Longitude} = \text{atan2}(\Delta\text{Longitude}, \text{denom});$
9. $\text{distanceToTarget} = \Delta\text{Longitude} \times 6,372,795;$
10. return distanceToTarget;

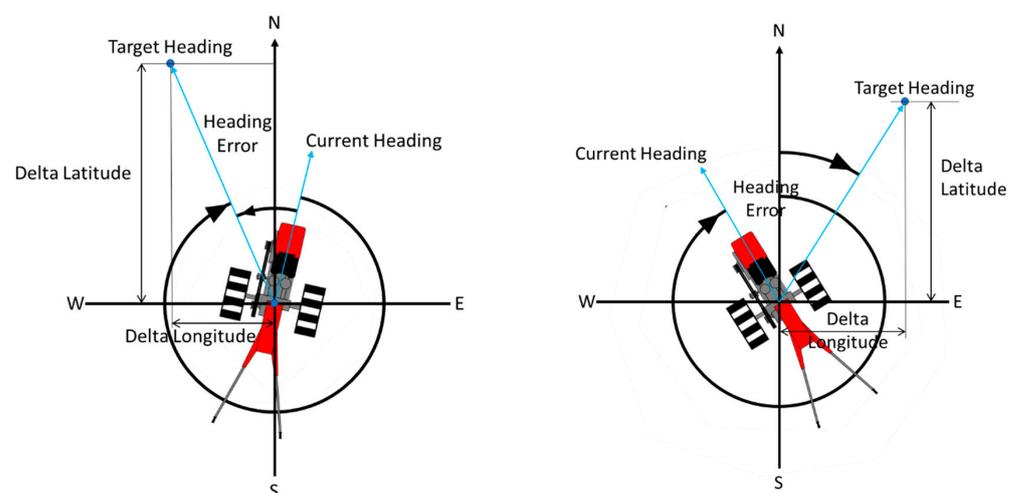
Algorithm 4: Calculate robot turning to get to the waypoint target.

Data: headingError, targetHeading, currentHeading, errorOutput, headingTolerance, motorMovement

Result: motorMovement

1. $\text{headingError} = \text{targetHeading} - \text{currentHeading};$
2. **if** (headingError less than -180) **then**
3. $\text{headingError} += 360;$
4. **else if** (headingError more than 180) **then**
5. $\text{headingError} -= 360;$
6. **end if**
7. $\text{errorOutput} \leftarrow \text{calcErrorOutput}(\text{headingError})$
8. $\text{motorMovement} \leftarrow \text{calcMotorMovement}(\text{errorOutput})$
9. return motorMovement;

Algorithm 4 is a heading error computation when the target heading value differs from the current heading value. The heading error calculation findings will be utilized as the input for the robot's movement, allowing it to operate autonomously. The direction of the tractor's heading is depicted in Figure 10. Two restrictions limit the results of the tractor heading angle calculation.



Case 1: Target Heading–Current Heading > 180

Case 2: Target Heading–Current Heading > -180

Figure 10. Tractor heading error.

Field tests were conducted and compared to simulation results to validate the Path Planning and Embedded System Platform. In this test, one scenario with the following characteristics was created:

1. Figure 11 depicts a waypoint’s coordinates with ROI, start-finish points, and four-meter interval distance. The value of four meters is utilized since the accuracy of the employed GPS is insufficient to overcome interval values of less than four meters.
2. Field test results are compared with the tractor model simulation using the MatLab robotic toolbox with a differential drive kinematic model to determine the estimated plowing time.
3. The simulation parameters used to represent the G1000 Quick tractor are shown in Table 3.
4. The ROI and the waypoint coordinates used in the simulation are the same as the size generated from the Path Planning Platform.

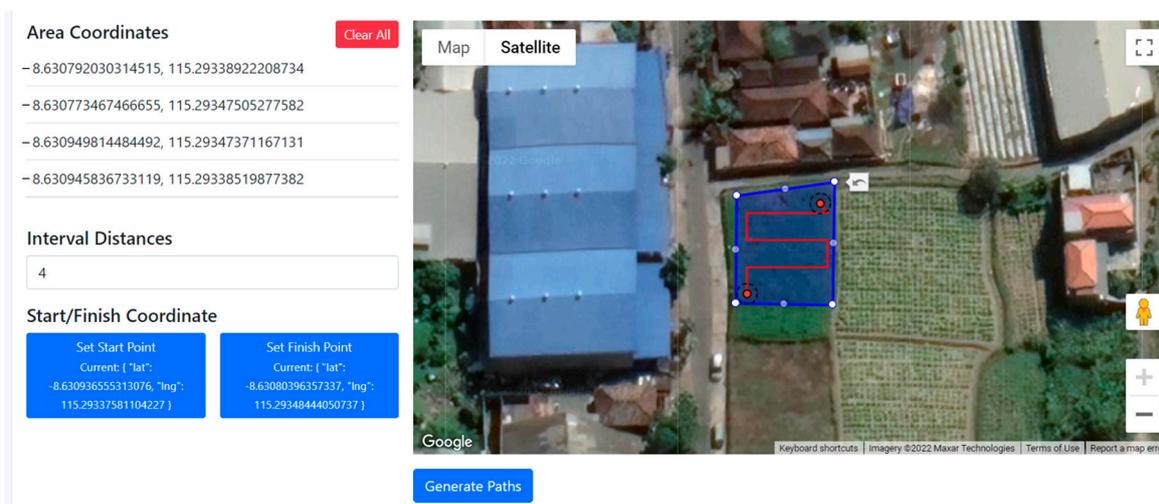


Figure 11. Field Trial Scenario with four-meter interval distance.

Table 3. Parameters for MatLab simulation.

Parameter	Unit	Value
Linear Velocity	m/s	0.56
Wheel Radius	rad/s	$\pi/4$
Distance between wheels	m	1.8

Figure 12 shows the Simulink block diagram for the simulation. The waypoint block is an input containing an array of coordinates. The experimental scenario of the waypoint block has an array measuring 8×2 . Based on the simulation results depicted in Figure 13, it is evident that the tractor follows the given coordinates of the path. This simulation shows that the estimated tractor movement time is 84 s. The simulation results are then compared with the tractor movement time in field trials, as shown in Table 3.

Field trials were carried out in one of the rice fields in the Ketewel area, Gianyar Regency, Bali, Indonesia. We used the same region of interest, starting point, finish point, and four meters distance interval. Each point is recorded using an IoT (Internet of Things) based logger module installed on the tractor. The data are stored in the database and processed using a text editor into gpx format, then visualized using the Google Earth application in Figure 14. The white line and red dots represent the robot’s target path and waypoints, while the yellow dots represent the tractor path and the recording results of the logger module. This experiment demonstrates how a robot can calculate tractor orientation and pass waypoints. As seen in Figure 14, the highest error distance between

the target waypoint and the tractor position based on the GPS and Kalman Filter is 2.61 m. This is due to the type of GPS module (U-Blox Neo-M8N) used, which has an accuracy of 2.5 m. We also recorded the time data of plowing carried out in simulations and field tests with an area of interest of 302.65 m², and the results are shown in Table 4. The time required to generate the path planning coordinates is 685 milliseconds, while the time between simulation and trial is different. There is a difference of a few seconds in which the simulation conditions are faster than the actual field trial.

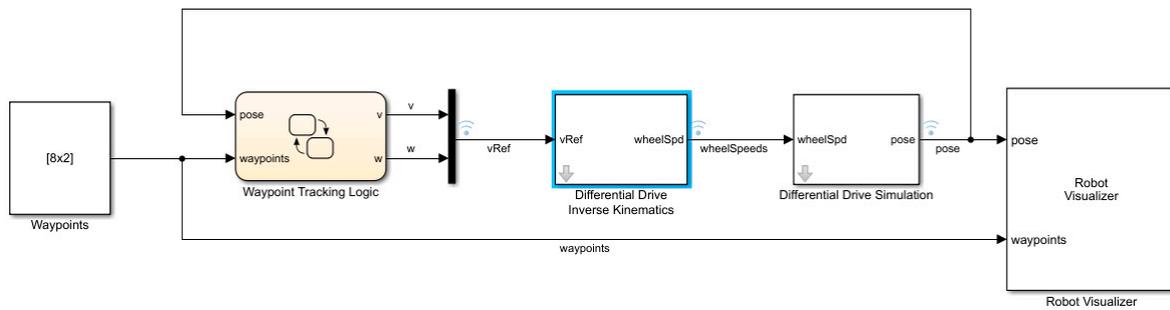


Figure 12. Matlab/Simulink waypoint simulation block diagram.

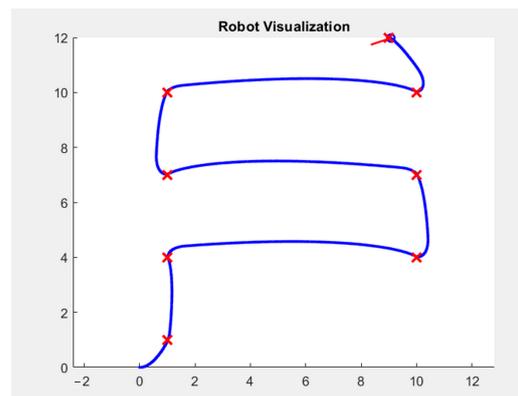


Figure 13. Robot movement simulation results. Visualization of robot movement using four meters distance between lines.



Figure 14. Data visualization on GPS navigation.

Table 4. Comparison of plowing process time between simulation results and field trials.

Polygonal ROI	Processing Time	Simulation Tillage Time	Tillage Time
Test 1	685 ms	84 s	132 s

5. Conclusions

A two-dimensional path planning platform using an edge-vertex path algorithm based on Laravel and Google Maps for the Autonomous Walk-Behind Hand Tractor has been presented. The algorithm automatically generates paths by considering the path interval distance (puddler width), start point, and finish point. Field trials were conducted with a walk-behind tractor to validate the resulting waypoint coordinates for rice field plowing missions. By applying the Kalman Filter, interference from GPS signals and magnetometer noise can be eliminated in certain instances. This system combines the Haversine formula and heading error calculation to automate waypoint navigation in tractors. The proposed algorithm permits farmers to specify the distance between paths based on varying puddler widths. Future research aims to design a path with a spiral pattern and to account for external disturbances, such as land contours, when constructing the path. In addition, the use of IoT technology is possible in the future because this path-planning platform can be implemented in the cloud.

Author Contributions: Conceptualization, P.N.C. and D.M.; methodology, P.N.C. and D.M.; software, P.N.C. and D.M.; validation, P.N.C. and D.M.; formal analysis, P.N.C. and D.M.; investigation, P.N.C. and D.M.; resources, P.N.C. and D.M.; data curation, P.N.C. and D.M.; writing—original draft preparation, P.N.C.; writing—review and editing, P.N.C. and D.M.; visualization, P.N.C.; supervision, D.M.; project administration, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We would like to express our deepest gratitude to the Ministry of Research, Technology and Higher Education of the Republic of Indonesia, the STIKOM Bali Institute of Technology, and the Business and Rajamangala University of Technology Thanyaburi (RMUTT) for the support and facilities that were provided for this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Castillejo, P.; Johansen, G.; Cürüklü, B.; Bilbao-Arechabala, S.; Fresco, R.; Martínez-Rodríguez, B.; Pomante, L.; Rusu, C.; Martínez-Ortega, J.F.; Centofanti, C.; et al. Aggregate Farming in the Cloud: The AFarCloud ECSEL Project. *Microprocess. Microsyst.* **2020**, *78*, 103218. [[CrossRef](#)]
- Almoaili, E.; Kurdi, H. Path Planning Algorithm for Unmanned Ground Vehicles (UGVs) in Known Static Environments. *Procedia Comput. Sci.* **2020**, *177*, 57–63. [[CrossRef](#)]
- Asadi, K.; Kalkunte Suresh, A.; Ender, A.; Gotad, S.; Maniyar, S.; Anand, S.; Noghabaei, M.; Han, K.; Lobaton, E.; Wu, T. An Integrated UGV-UAV System for Construction Site Data Collection. *Autom. Constr.* **2020**, *112*, 103068. [[CrossRef](#)]
- Qi, L.; Zhang, T.; Xu, K.; Pan, H.; Zhang, Z.; Yuan, Y. A Novel Terrain Adaptive Omni-Directional Unmanned Ground Vehicle for Underground Space Emergency: Design, Modeling and Tests. *Sustain. Cities Soc.* **2021**, *65*, 102621. [[CrossRef](#)]
- Radmanesh, M.; Sharma, B.; Kumar, M.; French, D. PDE Solution to UAV/UGV Trajectory Planning Problem by Spatio-Temporal Estimation during Wildfires. *Chin. J. Aeronaut.* **2021**, *34*, 601–616. [[CrossRef](#)]
- Fotio Tiotsop, L.; Servetti, A.; Masala, E. An Integer Linear Programming Model for Efficient Scheduling of UGV Tasks in Precision Agriculture under Human Supervision. *Comput. Oper. Res.* **2020**, *114*, 104826. [[CrossRef](#)]
- Galceran, E.; Carreras, M. A Survey on Coverage Path Planning for Robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [[CrossRef](#)]
- Arkin, E.M.; Fekete, S.P.; Mitchell, J.S.B. Approximation Algorithms for Lawn Mowing and Milling. *Comput. Geom.* **2000**, *17*, 25–50. [[CrossRef](#)]
- Huang, W.H. Optimal Line-Sweep-Based Decompositions for Coverage Algorithms. In Proceedings of the 2001 ICRA—IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, Republic of Korea, 21–26 May 2001.

10. Santos, L.C.; Santos, F.N.; Solteiro Pires, E.J.; Valente, A.; Costa, P.; Magalhaes, S. Path Planning for Ground Robots in Agriculture: A Short Review. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2020, Ponta Delgada, Portugal, 15–17 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 61–66.
11. Vasquez-Gomez, J.I.; Marciano-Melchor, M.; Valentin, L.; Herrera-Lozada, J.C. Coverage Path Planning for 2D Convex Regions. *J. Intell. Robot. Syst. Theory Appl.* **2020**, *97*, 81–94. [[CrossRef](#)]
12. Choset, H.; Pignon, P. Coverage Path Planning: The Boustrophedon Cellular Decomposition. In *Field and Service Robotics*; Springer: London, UK, 1998; pp. 203–209.
13. Moravec, H.P.; Elfes, A. High Resolution Maps from Wide Angle Sonar. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; IEEE: Piscataway, NJ, USA, 1985; pp. 116–121.
14. Zhu, D.; Tian, C.; Sun, B.; Luo, C. Complete Coverage Path Planning of Autonomous Underwater Vehicle Based on GBNN Algorithm. *J. Intell. Robot. Syst. Theory Appl.* **2019**, *94*, 237–249. [[CrossRef](#)]
15. Song, J.; Gupta, S. ϵ^* : An Online Coverage Path Planning Algorithm. *IEEE Trans. Robot.* **2018**, *34*, 526–533. [[CrossRef](#)]
16. Kapanoglu, M.; Alikalfa, M.; Ozkan, M.; Yazıcı, A.; Parlaktuna, O. A Pattern-Based Genetic Algorithm for Multi-Robot Coverage Path Planning Minimizing Completion Time. *J. Intell. Manuf.* **2012**, *23*, 1035–1045. [[CrossRef](#)]
17. An, V.; Qu, Z.; Crosby, F.; Roberts, R.; An, V. A Triangulation-Based Coverage Path Planning. *IEEE Trans. Syst. Man. Cybern. Syst.* **2020**, *50*, 2157–2169. [[CrossRef](#)]
18. Wei, M.; Isler, V. Coverage Path Planning under the Energy Constraint. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21 May 2018.
19. Li, Y.; Chen, H.; Joo Er, M.; Wang, X. Coverage Path Planning for UAVs Based on Enhanced Exact Cellular Decomposition Method. *Mechatronics* **2011**, *21*, 876–885. [[CrossRef](#)]
20. Ruiz-Larrea, A.; Roldán, J.J.; Garzón, M.; del Cerro, J.; Barrientos, A. A UGV Approach to Measure the Ground Properties of Greenhouses. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 418, pp. 3–13. ISBN 9783319271484.
21. Ohi, N.; Lassak, K.; Watson, R.; Strader, J.; Du, Y.; Yang, C.; Hedrick, G.; Nguyen, J.; Harper, S.; Reynolds, D.; et al. Design of an Autonomous Precision Pollination Robot. In Proceedings of the 2018 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7711–7718.
22. Zoto, J.; Musci, M.A.; Khaliq, A.; Chiaberge, M.; Aicardi, I. Automatic Path Planning for Unmanned Ground Vehicle Using UAV Imagery. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 980, pp. 223–230.
23. Rahman, M.M.; Ishii, K.; Noguchi, N. Optimum Harvesting Area of Convex and Concave Polygon Field for Path Planning of Robot Combine Harvester. *Intell. Serv. Robot* **2019**, *12*, 167–179. [[CrossRef](#)]
24. Crisnapati, P.N.; Maneetham, D. RIFIS: A Novel Rice Field Sidewalk Detection Dataset for Walk-Behind Hand Tractor. *Data* **2022**, *7*, 135. [[CrossRef](#)]
25. Crisnapati, P.N.; Maneetham, D.; Triandini, E. Trolls: A Novel Low-Cost Controlling System Platform for Walk-behind Tractor. *Int. J. Electr. Comput. Eng.* **2023**, *13*, 842. [[CrossRef](#)]
26. Preparata, F.P.; Shamos, M.I. *Computational Geometry: An Introduction*; Gries, D., Schneider, F., Eds.; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
27. Kumar, G.V.P.; Raheman, H. Development of a Walk-behind Type Hand Tractor Powered Vegetable Transplanter for Paper Pot Seedlings. *Biosyst. Eng.* **2011**, *110*, 189–197. [[CrossRef](#)]
28. Gan, H.; Lee, W.S. Development of a Navigation System for a Smart Farm. *IFAC-PapersOnLine* **2018**, *51*, 1–4. [[CrossRef](#)]
29. Berber, M.; Ustun, A.; Yetkin, M. Comparison of Accuracy of GPS Techniques. *Measurement* **2012**, *45*, 1742–1746. [[CrossRef](#)]
30. Karataş, G.B.; Karagoz, P.; Ayran, O. Trajectory Pattern Extraction and Anomaly Detection for Maritime Vessels. *Internet Things* **2021**, *16*, 100436. [[CrossRef](#)]
31. Fouché, G.J.; Malekian, R. Drone as an Autonomous Aerial Sensor System for Motion Planning. *Measurement* **2018**, *119*, 142–155. [[CrossRef](#)]