Int j simul model 22 (2023) 3, 392-403
Original scientific paper

DYNAMIC SAMPLING ALGORITHM FOR AGRICULTURE-MONITORING GROUND ROBOT

Yehoshua, A.[*]; Bechar, A.[*,**]; Cohen, Y.[**]; Shmuel, L.[**,***] & Edan, Y.[*]

[*] Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er Sheva 8410501, Israel

[**] Institute of Agricultural Engineering, Agricultural Research Organization, Bet Dagan 50250, Israel

[***] Department of Plant Pathology and Microbiology, Institute of Environmental Sciences, The Robert H. Smith Faculty of Agriculture, Food and Environment, The Hebrew University of Jerusalem, Rehovot, 7610001, Israel

E-Mail: adiyeho@post.bgu.ac.il

Abstract

We present the development and evaluation of a dynamic sampling algorithm for an agriculture-monitoring ground robot designed to locate insects in an agricultural field, where complete sampling of all plants is infeasible due to resource constraints. The algorithm utilizes real-time information to prioritise sampling at suspected points, locate hot spots and adapt sampling plans accordingly. A simulation environment was constructed to examine the algorithm's performance, and it was compared to two existing algorithms using Tetranychidae insect data from previous research. Sensitivity analyses reveals that the dynamic algorithm outperformed the others in all tested use cases, reaching 100 % detection approximately 3–5 days sooner when applied to small fields, and identifying 30 %–50 % more insects for larger fields. Its high detection percentages in small fields – 100 for a 1 ha field – decreased moderately with increasing field size to 80 % for a 10 ha field, seemingly irrespective of insect spread rate, which also barely affected insect detection. Doubling the time spent on each sample improved the results by 30–50 % on average in the first ten days, but in the following days the gap narrows.

(Received in February 2023, accepted in May 2023. This paper was with the authors 2 weeks for 1 revision.)

Key Words: Field Sampling, Agricultural Monitoring, Ground Robot, Dynamic Sampling Algorithm, Insect Detection

1. INTRODUCTION

Global food demand is projected to increase by 60 % and in developing countries by 100 %, in the next 25 years, due to the combined effects of increasing population and richer diet [1]. To feed the world's population, crop yield and production efficiency must be increased. Insect and plant diseases are major factors causing production and economic losses in the agricultural industry [2]. Direct yield losses caused by pathogens range between roughly 20 % and 40 % of global agricultural productivity [2]. Therefore, it is important to detect insects as early as possible because late detection further increases the negative effects of diseases on the quality and quantity of agricultural produce. While agricultural robots have been extensively researched and developed for a multitude of field operations [3], in the area of monitoring robots most of the research has concentrated on detection algorithms [4] and improving path planning [5]; there has been very little focus on improving cycle time [6]. In a pest inspection task, the cycle time consists of navigation, sensing, mapping and action operations [7]. A primary bottleneck in agriculture- monitoring robots is the travel time between plants [4]. The most common practice for carrying out the required task (e.g., spraying, detecting insects, monitoring) is to go through the entire field and sample all of the plants.

Several studies have focused on field-coverage planning to find an efficient route that minimises time and travel over the field [8]. Different algorithms have been applied, including greedy algorithms [9], the dragonfly algorithm [10], genetic algorithms [11], ant colony optimisation [12], and A* and dynamic A* (D*) algorithms [13]. All of them try to determine the optimal route to navigate in the field, which means trying to find an efficient order for plant sampling, but none suggest which plants to sample. The traveling salesman problem algorithm has been applied to save time while navigating between specific plant coordinates [14]. However, in most cases it is impractical; the robot cannot sample every plant in the field because of limited resources such as time or energy. When sampling all the plants in the field is not possible, an algorithm that will suggest which plants should be sampled is required.

Several robotic systems have been developed as data-gathering tools for environmental monitoring, allowing new perspectives and a greater understanding of environmental processes [15]. In contrast to traditional sensors that provide fixed monitoring points, robots can adapt to changes in the surrounding environment and can manipulate the sensors to the optimal position or to interact with objects in the environment to collect quality data. Extensive research has been conducted on the application of marine [16], terrestrial [17] and airborne robotic systems [18] for a variety of ecological monitoring tasks [15]. This includes oceanographic measurements [19], atmospheric conditions [20], dust [21], mapping [22], and monitoring of physical properties of the environment [23]. Despite these tremendous efforts, to the best of our knowledge, the literature on the development of robotic systems for insect detection and classification is sparse and focuses only on bee [24] or locust-swarm tracking [25].

The two-spotted spider mite of the family Tetranychidae is a cosmopolitan polyphagous pest, and is the most common mite in Israel [26]. To date, it has been found to feed on about 1300 host plants from 70 different botanical families [26]. This red mite is a pest species of high economic importance for crops growing in glasshouses and net houses, as well as for many other open crops [27]. Its wide global distribution greatly enhances the resultant damage [26]. Previous studies have revealed that its dissemination has a fixed pattern [27] with respect to the way it enters the field and spreads within it. The insects mainly spread from the field borders, and the plants that are next to the paths in the field. Plants located in the middle of the field are less likely to initiate the spread. Another insight from previous work [28] is the route of these insects' spread inside the field, over time: the denser the field (the more plants that are closer to each other), the higher the chances that the insects will spread to nearby plants and create hot spots (where there are large groups of insects); the more spacious the field, the slower the spread [29].

Modelling and simulation techniques are widely used in research for their advantages over traditional field tests [30], including risk-free experimentation, cost-effectiveness, repeatability, and ease of data collection [31, 32]. These benefits make simulation a valuable tool for researchers, allowing them to study complex systems, and make informed decisions. In agriculture systems the need for simulation is amplified. The introduction of simulation into agricultural systems provides the ability to compare several alternatives under predefined, controlled conditions that are independent of the growing season, without the need for repeated field experiments [33] which are tedious and time consuming. The influence of differences between and within cultivars and growing conditions can be examined with a computerized model of the system [34], and statistical comparisons can be made among the various possible combinations of all crop parameters, such as the crop geometry and the fruit distribution [35].

A very important aspect for the operation of an insect-detecting robot traversing a field with limited time is the ability to perform the sampling in a dynamic manner, based on prior knowledge and real-time information. Thus, knowing the patterns that characterise insects' entrance into the field or spread will allow us to choose better sampling points. In addition, use of real-time information will allow the robot to change its sampling route, while in motion, to locate new areas where insects spread. Our aim was to develop a dynamic sampling algorithm which will allow the robot to sample the field at strategic points based on prior knowledge, and through them to locate areas of insect spread and further sampling. This algorithm should strive to find the hot spots while reducing the number of sampling points, and be able to dynamically change its plans accordingly to ensure accurate spatial and temporal sampling of the insects.

2. METHOD

2.1 Overview

A simulation environment was developed to evaluate the performance of the dynamic sampling algorithm in several different scenarios, including extreme cases. The simulation settings were arranged to resemble an insect-infested field. In order to test the capabilities of the dynamic sampling algorithm, it was compared to two others. The comparison was conducted according to three use cases derived from insect data collected in parallel research. In addition, a sensitivity analysis was carried out to evaluate how the algorithm will react in different situations.

2.2 Algorithms

The dynamic sampling algorithm was compared to the ‘Naive’ and ‘Bouncy’ algorithms.

The leading principle of the dynamic sampling algorithm (Fig. 1) is to primarily focus on the areas where it is more likely to find an insect, based on previous studies that indicates a fixed pattern of the insect distribution. Insects primarily spread from field borders and plants near paths, while those in the middle of the field have a lower likelihood of initiating spread. When a positive detection occurs, the algorithm prioritizes nearby plants to efficiently locate hot spots. If the detection result is positive at a specific location, greater emphasis is placed on sampling nearby plants in an attempt to identify large areas of spread (hot spots) thoroughly and efficiently.

Initially, two lists are created: an 'Open list' for plants planned to be visited, where the robot selects the closest plant for sampling, and a 'Close list' for visited plants. A plant is inserted into the 'Open list' if it meets the criteria of being suspicious, which is determined based on real-time data (such as detecting an insect on a nearby plant) and prior information such as insect-spread patterns. Whenever an insect is detected, the robot adds nearby plants to the 'Open list'. The sampling loop continues until the end of the workday or until the 'Open list' is empty. When the list becomes empty, unvisited suspicious plants are then added to the list.

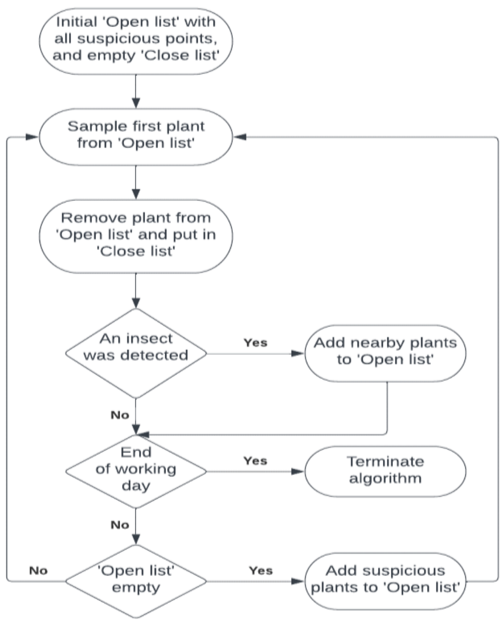

Figure 1: Dynamic sampling algorithm flow diagram.

In the ‘Naive’ algorithm, the robot samples every plant in the field, one by one, in a predetermined order. In the ‘Bouncy’ algorithm, the robot samples every N plants in the field in a predetermined order (for example, when $N = 2$, half of the plants in the field will be sampled). Throughout this article, the 'Bouncy' algorithm was used with an N value of 2.

2.3 Environment

A virtual environment was developed in the Gazebo simulator, a widely-used robotics simulation tool known for its realistic physics engine and customizable 3D environment. The robot's motion is visualized in RViz software, providing a comprehensive understanding of the environment without the need for physical fieldwork or real insects. This simulation setup enables seamless transfer of algorithms to an actual robot system. Fig. 2 showcases the robot in the Gazebo and RViz environments, highlighting the integrated simulation and visualization tools.

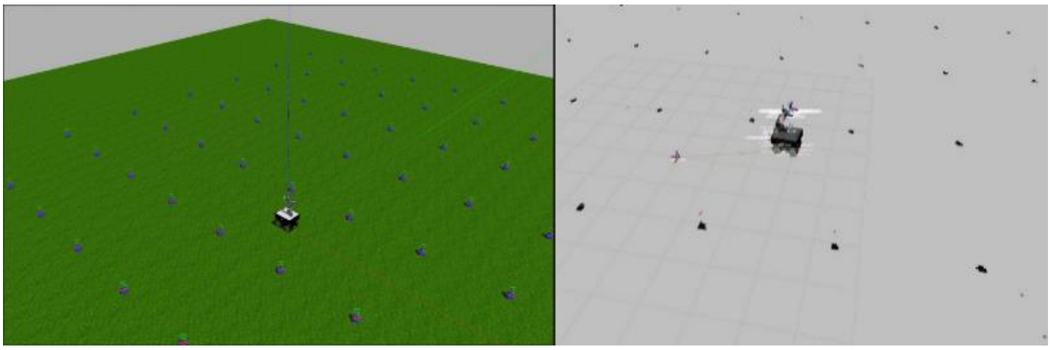

Figure 2: Gazebo (left) and RViz (right) environments.

2.4 Assumptions and limitations

The following assumptions were considered when developing the simulation model:

- Each grower has only one robot, for a defined duration each day (working day).
- The paths are free of obstacles.
- The robot travels along each row and samples the plants.
- Whenever a row ends, the robot moves forward to the next row.
- The robot cannot cross rows and must drive to the end of each row to move on.
- To perform the inspection, the robot stops next to the plant that is being sampled.
- When the robot reaches a plant, it inspects the plant from several viewpoints in order to detect the insects; the more viewpoints sampled, the longer the robot stays next to the plant. The addition of inspection viewpoints increases the inspection detection performance (reliability).
- If an insect is detected, it is destroyed and does not spread any further.
- Plants can be sampled only once each day, but multiple times during the simulation (along different days).
- If an insect is not detected, it can spread to adjacent plants, whether along the same row or across to the next row, at the end of the day.
- The robot might miss an insect during inspection.
- In cases where we analyse the influence of different sampling times and detection rates, values for sampling times and detection rates were used as detailed below:
    - When the time spent searching for an insect is less than 20 s (less than one viewpoint), the detection rate will be 0 %.
    - When the time spent searching for an insect is 20 s (one viewpoint), the detection rate will be 50 %.

- When the time spent searching for an insect is 25 s (two viewpoints), the detection rate will be 60 %.
- When the time spent searching for an insect is 30 s (three viewpoints), the detection rate will be 69 %.
- When the time spent searching for an insect is 35 s (four viewpoints), the detection rate will be 77 %.
- When the time spent searching for an insect is 40 s (five viewpoints), the detection rate will be 84 %.
- When the time spent searching for an insect is more than 40 s (more than five viewpoints), the detection rate will be 90 %.

2.5 Parameters

The following independent parameters were used in the simulation as default parameters:
- The robot works 25 days, 12 h per day.
- The robot's average speed is 0.7 m/s.
- The distance between adjacent rows is 4 m with a 3 m distance between plants.
- The robot spends 40 s next to each plant that it samples, and the detection rate is 84 %.

2.6 Analyses

Three different use cases were conducted to compare the three algorithms. The use cases were formulated to focus on multiple parameters as follows:
- The size of the field (1 ha, 5 ha, 10 ha).
- Number of plants in the field.

The use cases were based on data collected in previous research [33]. Two-spotted spider mite data were based on a dataset monitored in Paran, Israel [33], which was collected once a week from five different plots during the growing seasons of 2015–2017, and the pest's spreading behaviour was tracked using GPS-GIS technologies [34]. The simulation of the robot's traversal through the field, sampling the plants and trying to detect insects, was repeated 10 times for each scenario to ensure robustness and reliability of the results. The presented results are the average of all repetitions.

<u>Scenario A</u> – Small plot size of 1 ha and 784 plants in the field.

<u>Scenario B</u> – Medium plot size of 5 ha and 4096 plants in the field.

<u>Scenario C</u> – Large plot size of 10 ha and 8281 plants in the field.

Sensitivity analysis was performed for the following parameters: plot sizes (from 1 to 20 ha), spread rates each day (30, 50 and 80 %), detection times (20, 25, 30, 35 and 40 s) and robot detection rates (50, 60, 70, 80 and 90 %).

The following performance measures were calculated for each day and cumulatively until the end of the working days:
- Percent detection – the percentage of insects located.
- Percent plants visited – the percentage of plants sampled.

3. RESULTS AND DISCUSSION

3.1 Algorithm comparisons

For Scenario A, all three algorithms reached 100 % detection at the end of the working days, with the goal being reached after 2 days for both the dynamic sampling and 'Naive' algorithms and after 4 days for the 'Bouncy' algorithm (Fig. 3, left side).

Right side of Fig. 3 presents the percentage of plants out of all of the plants in the field that were sampled by the robot, each day. As expected, the more plants the robot sampled on each working day, the faster the algorithm reached 100 % detection. The ‘Naïve’ algorithm, which sampled the largest number of plants each working day (an average of 67 % of the plants), and the dynamic sampling algorithm (an average of 53 %) achieved 100 % detection after 2 days of work, whereas for the ‘Bouncy’ algorithm (an average of 47 % of the plants each day), it took 4 days to discover all of the insects in the field.

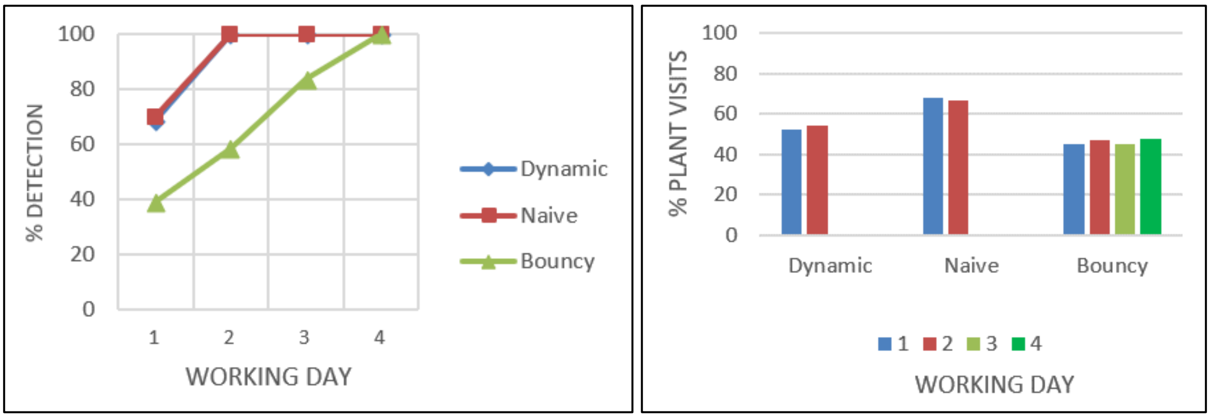

Figure 3: Scenario A – percent detection (left) and percent plants visited (right).

For Scenario B, there was a significant difference between the algorithms in percent detection, with the dynamic sampling algorithm achieving 84 % detection at the end of the working days, compared to 68 % for the ‘Naïve’ algorithm and 46 % for the ‘Bouncy’ algorithm (Fig. 4, left side).

Analysis of the percentage of plants out of all of the plants in the field that were sampled by the robot, each day (Fig. 6, right side) revealed an interesting insight: it is not necessary to sample many plants every day to ensure detection of a large number of insects. Although the dynamic sampling algorithm achieved the most rapid and best results, it was not the one that sampled the largest number of plants each day. On the other hand, while the ‘Naive’ algorithm succeeded in sampling a larger number of plants during each working day separately (average of 15 %), it gave worse results than the dynamic sampling algorithm (average of 12 %). The shorter the distance travelled by the robot between samples, the less time it spends travelling in the fields and the more time it devotes to sampling, thus managing to reach a larger number of plants. Therefore, the number of visits with the dynamic sampling algorithm on the first days was more similar to that with the 'Naive' algorithm. For every sample in which an insect is detected with the dynamic sampling algorithm, the robot will not skip plants, but as the days pass, and the number of insects in a field decrease, the robot will skip plants in order to locate hot spots and will behave more like the 'Bouncy' algorithm.

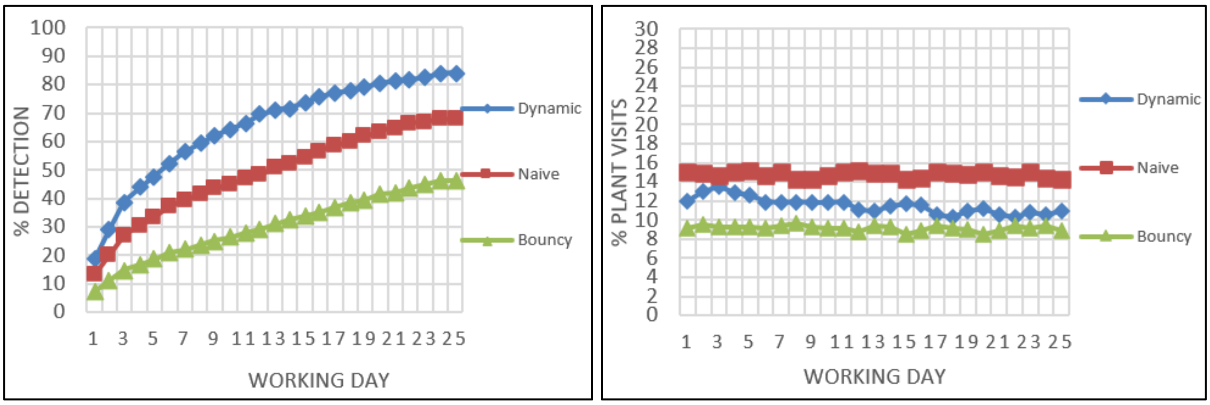

Figure 4: Scenario B – percent detection (left) and percent plants visited (right).

Results for Scenario C revealed a significant difference between the three algorithms in percent detection (Fig. 5). The 'Bouncy' algorithm gave the worst results (maximum 22 % detection) due to the low number of plants sampled each day, even though on the first days it showed very close performance to the 'Naive' algorithm. In contrast, the 'Naive' algorithm showed better results and consistent improvement over time, but at the end of the working days reached only 34 % detection. The dynamic sampling algorithm achieved the best results with 49 % detection at the end of the working days.

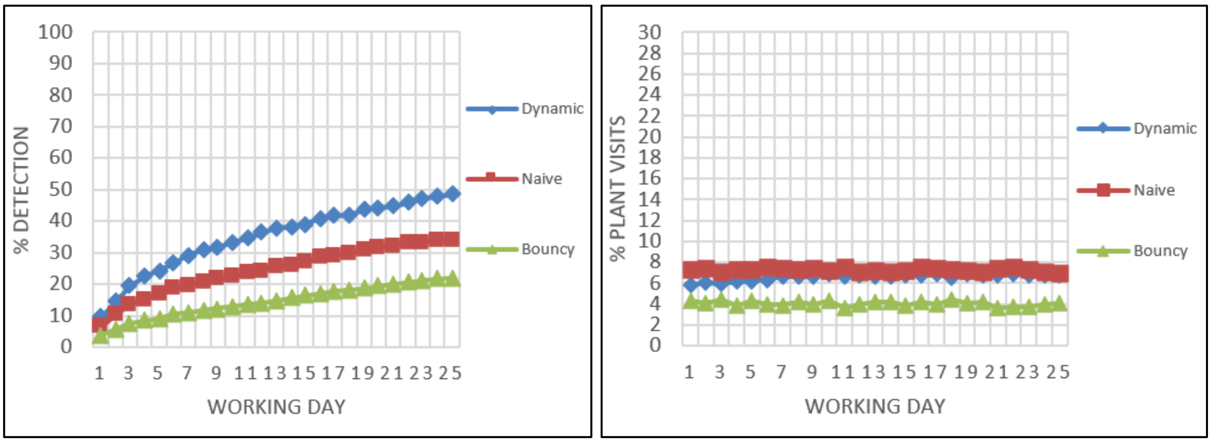

Figure 5: Scenario C – percent detection (left) and percent plants visited (right).

3.2 Sensitivity analysis

For the default parameter values, the dynamic sampling algorithm already produced excellent detection results of 100 % after about a week in a 1 ha field (Fig. 6). For most regular-sized fields (2–5 ha), it reached up to 90 % detection. In large-size fields, the robot failed to detect the same number of insects as in the smaller fields, and as the field size increased, there was a moderate decrease in the percentage of detection at the end of the working days.

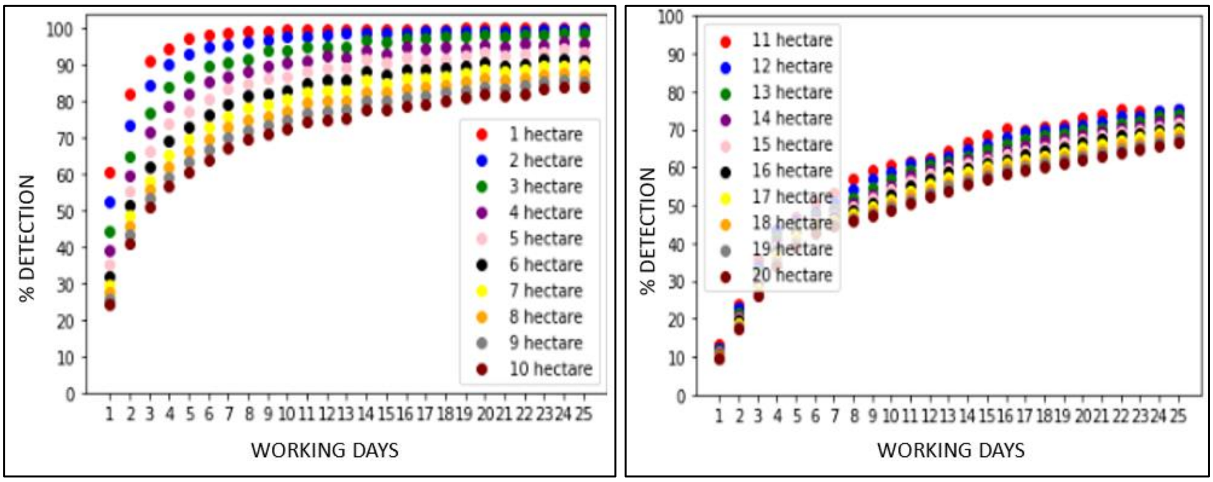

Figure 6: Percent detection in fields of different sizes.

As expected, as the size of the field increased, the detection percentage decreased (Fig. 7). The robot achieved a high detection result (90–100 %) in small-sized fields, but the percentages decreased sharply in fields between 7 ha and 14 ha. In larger fields, detection percentage was characterised by a moderate linear decrease.

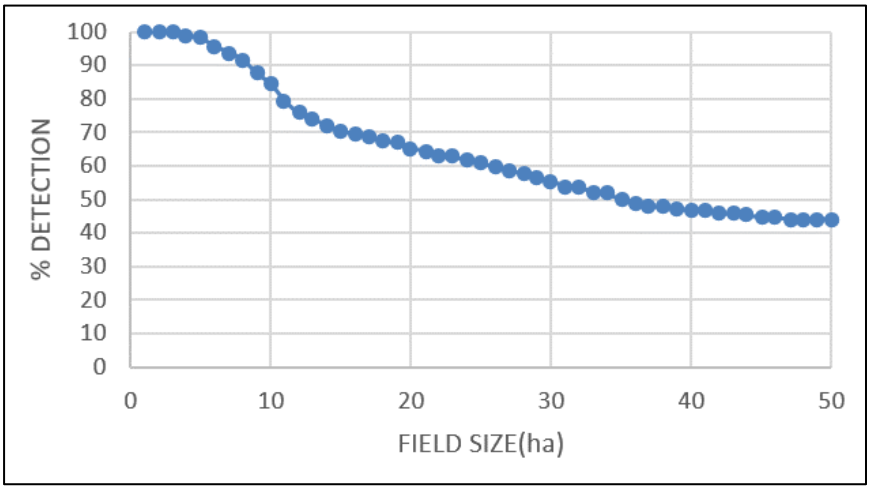

Figure 7: Percent detection vs. field size.

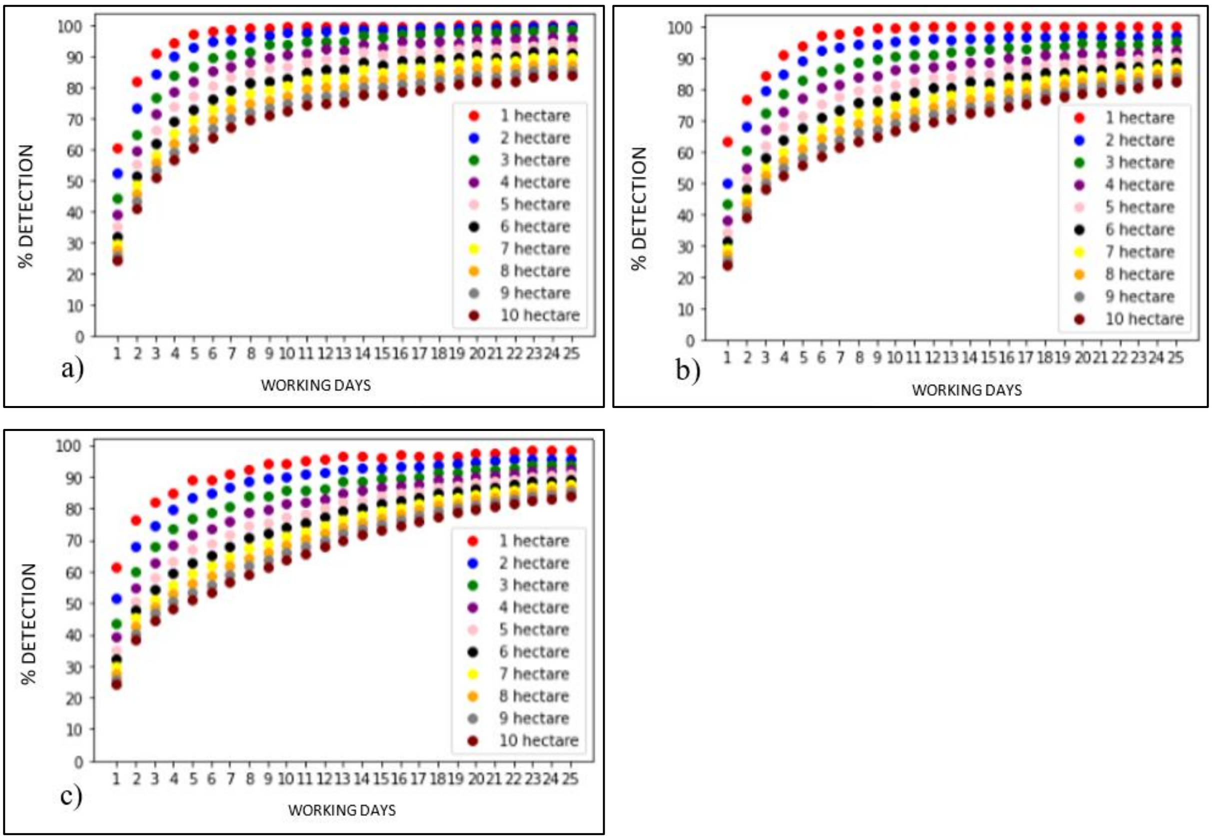

Figure 8: Percent detection for different spread rates: a) 30 %, b) 50 %, c) 80 %.

The insects’ rate of spread did not influence the final detection rates (Fig. 8). Although on the first few days there was a minor difference between the three rates of insect spread, where a lower rate of spread produced an average of 5 % more detection, the results of the robot were the same at the end of the working days for each of the different field sizes. This stems from the way in which the algorithm works, i.e., locating hot spots in the field and acting accordingly.

If an insect exists on each plant, the more time the robot invests searching for it, the greater the chance that it will find it. The trade-off between time spent searching for insects in each plant and the detection percentage was also tested (Fig. 9). Results revealed that for a field of 6 ha, even if the number of plants that are inspected each day is small, the more time the robot spends searching for insects in each plant, the more insects will be detected. Thus, already on the first day, the robot managed to detect 50 % more insects if it doubled the time spent on each plant (32 % in 40 s versus 21 % in 20 s). On the other hand, with time, the differences became smaller, and at the end of the working days, all cases reached over 90 % detection.

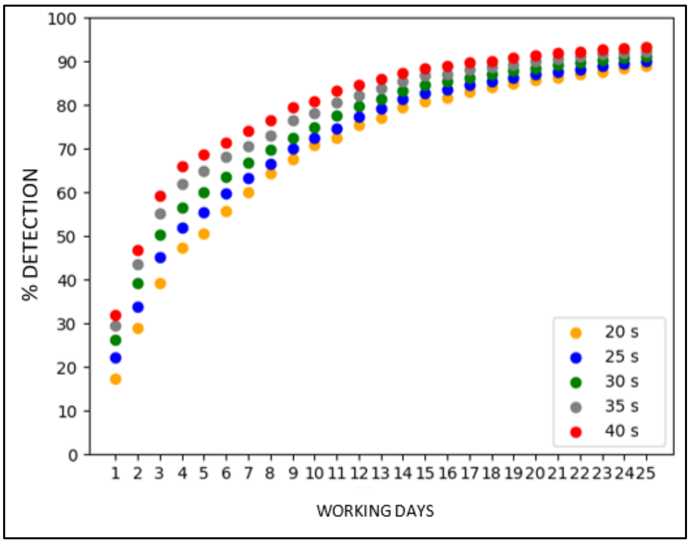

Figure 9: Percent detection for different visit times.

Fig. 10 presents the results for robots with different detection rates, where different robots (or the same one, with different detection capabilities) use the dynamic sampling algorithm. As expected, better detection capabilities led to better detection results. However, the dynamic sampling algorithm allowed up to 75 % detection, even for robots with a very low percentage of detection ability (40 %).

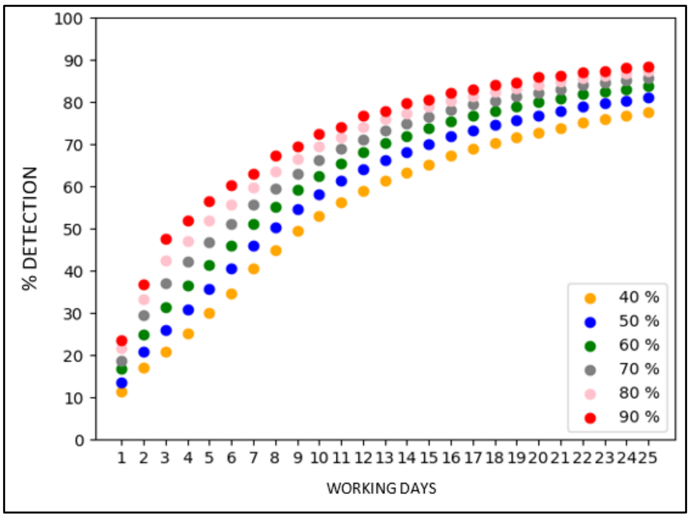

Figure 10: Percent detection – for robots with different detection rates.

4. CONCLUSION

A dynamic sampling algorithm for monitoring insects in agricultural fields under limited resources – time, money or energy, among others – is proposed to determine an efficient way for a mobile ground robot to sample an insect-infested field. The dynamic sampling algorithm uses real-time data from the field and decides where to sample based on a-priori information and on the plants that it samples along the way. By applying this approach, it is possible to increase detection rates, regardless of field size and robot detection capability.

The comparison of the dynamic sampling algorithm to two other existing algorithms, on real data collected in previous studies, revealed its better performance. Although for a small field of 1 ha, its result was similar to that of the 'Naive' algorithm, in more complex cases of larger fields or higher rates of insect spread, it presented up to 50 % better results for the detection of insects in the field.

In the sensitivity analyses, the dynamic sampling algorithm was tested against different rates of insect spread (simulating different insects) and presented identical results, regardless of the spread rate. The detection capabilities of the robot were also tested. Although on the early working days, it was evident that the better the robot's detection capabilities, the more effective the algorithm would be, over time, it was proven that the robot's capability was not an influencing factor; at the end of the simulation, the algorithm showed similar results, regardless of how good the robot's insect-detecting capability was. These comparisons would have been impossible without the use of simulation.

We therefore recommend utilization of the proposed dynamic sampling algorithm in agricultural monitoring applications as it demonstrates great performance in terms of detection rate and time efficiency when compared to existing algorithms, and its results are robust to variations in field size, insect spread rate and detection times. Real-world implementation on an actual mobile robotic system is underway.

ACKNOWLEDGEMENTS

This project was partially supported by the Ministry of Science Fund No. 3-16143 and Ministry of Agriculture Chief Scientist Fund No. 16-29-0004 and by Ben-Gurion University of the Negev through the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering and The Agricultural, Biological Cognitive Robotics Initiative.

REFERENCES

[1] Maxted, N.; Amri, A.; Castañeda-Álvarez, N. P.; Dias, S.; Dulloo, M. E.; Fielder, H.; Ford-Lloyd, B. V.; Iriondo, J. M.; Brehm, J. M.; Nilsen, L. B.; Thormann, I.; Vincent, H.; Kell, S. P. (2016). Joining up the dots: a systematic perspective of crop wild relative conservation and use, Maxted, N.; Dulloo, M. E.; Ford-Lloyd, B. V. (Eds.), *Enhancing Crop Genepool Use: Capturing Wild Relative and Landrace Diversity for Crop Improvement*, CABI, Wallingford, 87-124, doi:10.1079/9781780646138.0087

[2] Savary, S.; Ficke, A.; Aubertot, J.-N.; Hollier, C. (2012). Crop losses due to diseases and their implications for global food production losses and food security, *Food Security*, Vol. 4, No. 4, 519-537, doi:10.1007/s12571-012-0200-5

[3] Fountas, S.; Mylonas, N.; Malounas, I.; Rodias, E.; Hellmann Santos, C.; Pekkeriet, E. (2020). Agricultural robotics for field operations, *Sensors*, Vol. 20, No. 9, Paper 2672, 27 pages, doi:10.3390/s20092672

[4] Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. (2022). Applications of machine vision in agricultural robot navigation: a review, *Computers and Electronics in Agriculture*, Vol. 198, Paper 107085, 13 pages, doi:10.1016/j.compag.2022.107085

[5] Santos, L. C.; Santos, F. N.; Pires, E. S.; Valente, A.; Costa, P.; Magalhães, S. (2020). Path planning for ground robots in agriculture: a short review, *2020 IEEE International Conference on Autonomous Robot Systems and Competitions*, 61-66, doi:10.1109/ICARSC49921.2020.9096177

[6] Kurtser, P.; Edan, Y. (2020). Planning the sequence of tasks for harvesting robots, *Robotics and Autonomous Systems*, Vol. 131, Paper 103591, 12 pages, doi:10.1016/j.robot.2020.103591

[7] Botta, A.; Cavallone, P.; Baglieri, L.; Colucci, G.; Tagliavini, L.; Quaglia, G. (2022). A review of robots, perception, and tasks in precision agriculture, *Applied Mechanics*, Vol. 3, No. 3, 830-854, doi:10.3390/applmech3030049

[8] Hameed, I. A.; Bochtis, D.; Sørensen, C. A. (2013). An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas, *International Journal of Advanced Robotic Systems*, Vol. 10, No. 5, Paper 231, 9 pages, doi:10.5772/56248
[9] Oksanen, T.; Visala, A. (2009). Coverage path planning algorithms for agricultural field machines, *Journal of Field Robotics*, Vol. 26, No. 8, 651-668, doi:10.1002/rob.20300
[10] Muthukumaran, S.; Sivaramakrishnan, R. (2019). Optimal path planning for an autonomous mobile robot using dragonfly algorithm, *International Journal of Simulation Modelling*, Vol. 18, No. 3, 397-407, doi:10.2507/IJSIMM18(3)474
[11] Manikas, T. W.; Ashenayi, K.; Wainwright, R. L. (2007). Genetic algorithms for autonomous robot navigation, *IEEE Instrumentation & Measurement Magazine*, Vol. 10, No. 6, 26-31, doi:10.1109/MIM.2007.4428579
[12] Akka, K.; Khaber, F. (2018). Mobile robot path planning using an improved ant colony optimization, *International Journal of Advanced Robotic Systems*, Vol. 15, No. 3, 7 pages, doi:10.1177/1729881418774673
[13] Edan, Y.; Flash, T.; Peiper, U. M.; Shmulevich, I.; Sarig, Y. (1991). Near-minimum-time task planning for fruit-picking robots, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, 48-56, doi:10.1109/70.68069
[14] Kyaw, P. T.; Paing, A.; Thu, T. T.; Mohan, R. E.; Le, A. V.; Veerajagadheswar, P. (2020). Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem, *IEEE Access*, Vol. 8, 225945-225956, doi:10.1109/ACCESS.2020.3045027
[15] Dunbabin, M.; Marques, L. (2012). Robots for environmental monitoring: significant advancements and applications, *IEEE Robotics & Automation Magazine*, Vol. 19, No. 1, 24-39, doi:10.1109/MRA.2011.2181683
[16] Smale, D. A.; Langlois, T. J.; Kendrick, G. A.; Meeuwig, J. J.; Harvey, E. S. (2011). From fronds to fish: the use of indicators for ecological monitoring in marine benthic ecosystems, with case studies from temperate Western Australia, *Reviews in Fish Biology and Fisheries*, Vol. 21, No. 3, 311-337, doi:10.1007/s11160-010-9173-7
[17] Worsfold, P. J.; Clough, R.; Lohan, M. C.; Monbet, P.; Ellis, P. S.; Quétel, C. R.; McKelvie, I. D. (2013). Flow injection analysis as a tool for enhancing oceanographic nutrient measurements – a review, *Analytica Chimica Acta*, Vol. 803, 15-40, doi:10.1016/j.aca.2013.06.015
[18] He, X.; Bourne, J. R.; Steiner, J. A.; Mortensen, C.; Hoffman, K. C.; Dudley, C. J.; Rogers, B.; Cropek, D. M.; Leang, K. K. (2019). Autonomous chemical-sensing aerial robot for urban/suburban environmental monitoring, *IEEE Systems Journal*, Vol. 13, No. 3, 3524-3535, doi:10.1109/JSYST.2019.2905807
[19] Nicholls, K. W.; Abrahamsen, E. P.; Buck, J. J. H.; Dodd, P. A.; Goldblatt, C.; Griffiths, G.; Heywood, K. J.; Hughes, N. E.; Kaletzky, A.; Lane-Serff, G. F.; McPhail, S. D.; Millard, N. W.; Oliver, K. I. C.; Perrett, J.; Price, M. R.; Pudsey, C. J.; Saw, K.; Stansfield, K.; Stott, M. J.; Wadhams, P.; Webb, A. T.; Wilkinson, J. P. (2006). Measurements beneath an Antarctic ice shelf using an autonomous underwater vehicle, *Geophysical Research Letters*, Vol. 33, No. 8, Paper L08612, 4 pages, doi:10.1029/2006GL025998
[20] Chang, S.; Jeong, K. (2017). A mobile application for fine dust monitoring system, *2017 18th IEEE International Conference on Mobile Data Management*, 336-339, doi:10.1109/MDM.2017.55
[21] Reggente, M.; Lilienthal, A. J. (2010). The 3D-Kernel DM+V/W algorithm: using wind information in three dimensional gas distribution modelling with a mobile robot, *Proceedings of the 2010 IEEE Sensors*, 999-1004, doi:10.1109/ICSENS.2010.5690924
[22] Bryson, M.; Reid, A.; Ramos, F.; Sukkarieh, S. (2010). Airborne vision-based mapping and classification of large farmland environments, *Journal of Field Robotics*, Vol. 27, No. 5, 632-655, doi:10.1002/rob.20343
[23] SanClements, M. D.; Fernandez, I. J.; Norton, S. A. (2010). Soil chemical and physical properties at the Bear Brook Watershed in Maine, USA, *Environmental Monitoring and Assessment*, Vol. 171, No. 1-4, 111-128, doi:10.1007/s10661-010-1531-3
[24] Hart, N. H.; Huang, L. (2014). Monitoring populations of solitary bees using image processing techniques, *International Journal of Computer Applications in Technology*, Vol. 50, No. 1-2, 45-50, doi:10.1504/IJCAT.2014.063907

[25] Tahir, N.; Brooker, G. (2009). Feasibility of UAV based optical tracker for tracking Australian plague locust, *Australasian Conference on Robotics and Automation*, Paper 9, 10 pages

[26] Migeon, A.; Nouguier, E.; Dorkeld, F. (2010). Spider Mites Web: a comprehensive database for the Tetranychidae, Sabelis, M.; Bruin, J. (Eds.), *Trends in Acarology*, Springer, Dordrecht, 557-560, doi:10.1007/978-90-481-9837-5_96

[27] Gerson, U.; Weintraub, P. G. (2012). Mites (Acari) as a factor in greenhouse management, *Annual Review of Entomology*, Vol. 57, No. 1, 229-247, doi:10.1146/annurev-ento-120710-100639

[28] Shmuel, L. (2018). *Dynamics of the Two Spotted Spider Mites (Tetranychus Urticae) in Pepper Greenhouses and Characterization of Spatio-temporal Monitoring*, Master Thesis, Hebrew University of Jerusalem, Jerusalem

[29] Cieniewicz, E. J.; Pethybridge, S. J.; Gorny, A.; Madden, L. V.; McLane, H.; Perry, K. L.; Fuchs, M. (2017). Spatiotemporal spread of grapevine red blotch-associated virus in a California vineyard, *Virus Research*, Vol. 241, 156-162, doi:10.1016/j.virusres.2017.03.020

[30] Yan, L. X.; Jia, L.; Guo, J. H.; Lu, S. (2022). A simulation study on the identification of eco driving behaviour, *International Journal of Simulation Modelling*, Vol. 21, No. 3, 489-500, doi:10.2507/IJSIMM21-3-CO11

[31] Freile, A. J.; Mula, J.; Campuzano-Bolarin, F. (2020). Integrating inventory and transport capacity planning in a food supply chain, *International Journal of Simulation Modelling*, Vol. 19, No. 3, 434-445, doi:10.2507/IJSIMM19-3-523

[32] Zhao, X. F.; Liu, H. Z.; Lin, S. X.; Chen, Y. K. (2020). Design and implementation of a multiple AGV scheduling algorithm for a job-shop, *International Journal of Simulation Modelling*, Vol. 19, No. 1, 134-145, doi:10.2507/IJSIMM19-1-CO2

[33] Edan, Y.; Miles, G. E. (1994). Systems engineering of agricultural robot design, *IEEE transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 8, 1259-1265, doi:10.1109/21.299707

[34] Kurtser, P.; Edan, Y. (2020). Planning the sequence of tasks for harvesting robots, *Robotics and Autonomous Systems*, Vol. 131, Paper 103591, 12 pages, doi:10.1016/j.robot.2020.103591

[35] Kurtser, P.; Edan, Y. (2018). Statistical models for fruit detectability: spatial and temporal analyses of sweet peppers, *Biosystems Engineering*, Vol. 171, 272-289, doi:10.1016/j.biosystemseng.2018.04.017