Intelligent Insect–Computer Hybrid Robot: Installing Innate Obstacle Negotiation and Onboard Human Detection onto Cyborg Insect

*Phuoc Thanh Tran-Ngoc, Duc Long Le, Bing Sheng Chong, Huu Duoc Nguyen, Van Than Dung, Feng Cao, Yao Li, Kazuki Kai, Jia Hui Gan, Tat Thang Vo-Doan, Thanh Luan Nguyen, and Hirotaka Sato**

Developing small mobile robots for Urban Search and Rescue (USAR) is a major challenge due to constraints in size and power required to perform vital functions such as obstacle navigation, victim detection, and wireless communication. Drawing upon the idea that insects' locomotion can be controlled, what if we further utilize the insects' intrinsic ability to avoid obstacles? Herein, a cockroach hybrid robot (≈ 1.5 cm height, 5.7 cm length) that implements the abovementioned functions is developed. It is tested in an arena with randomly placed obstacles, and a motion capture system is used to track the insect's position among the untracked obstacles. A navigation algorithm that uses an inertial measurement unit (IMU) is developed to heuristically predict the insect's situation and stimulate the insect to escape nearby obstacles. The utilization of insect's intrinsic locomotor ability and low-powered IMU reduces the onboard power load, allowing the addition of a human-detecting function. An image classification model enables the use of an onboard low-resolution infrared camera for human detection. Consequently, a single hybrid robot is established that includes locomotion control, autonomous navigation in obstructed areas, onboard human detection, and wireless communication, representing a significant step toward real USAR application.

1. Introduction

In an Urban Search and Rescue (USAR) mission, fast pinpointing of the locations of victims trapped by structural collapse is crucial to maximize the chance of survival, as the initial reconnaissance work could take 3 h on average.[1] Since the locations of the trapped victims are usually inaccessible to humans, various tools such as electronic cameras, seismic sensors, and sniffing dogs are utilized to track down possible hidden victims in a small area. However, these methods have constraints such as handling difficulty, limited range, sensitivity to environment, and slow searching speed, which limits the number of victims that can be quickly located.[2] Mini robots with dimensions of 10 cm or less can be immensely helpful in increasing search efficiency, especially as they are compact and lightweight enough to penetrate through the rubble to locate victims. However, there are certain requirements for these types of robots to be used in search and rescue. First, they must have a high level of mobility to autonomously negotiate and traverse through obstacles without commands from a remote operator. They must also be able to detect life signatures of a human to locate survivors automatically. Finally, wireless communication is also necessary for them to report to a remote operator or a rescue team base.[2]

Owing to breakthroughs in fabrication technologies and actuator developments, researchers have built mini robots that can walk/crawl,[3–5] climb,[6] or fly,[7,8] which are compatible with

P. T. Tran-Ngoc, D. L. Le, B. S. Chong, H. D. Nguyen, V. T. Dung, F. Cao, K. Kai, J. H. Gan, H. Sato
School of Mechanical & Aerospace Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798, Singapore
E-mail: hirosato@ntu.edu.sg

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202200319.

DOI: 10.1002/aisy.202200319

Y. Li
School of Mechanical Engineering and Automation
Harbin Institute of Technology, Shenzhen
University Town, Shenzhen 518055, China

T. T. Vo-Doan
Institute of Biology I
University of Freiburg
Hauptstrasse 1, 79104 Freiburg, Germany

T. L. Nguyen
School of Computing
University of Leeds
Woodhouse, Leeds LS2 9JT, UK

USAR missions. State-of-the-art technology has made mini-insect-scale robots that crawl or fly in untethered conditions.[7,9–12] However, to the best of our knowledge, none of them are capable of autonomous navigation inside an unknown environment. This is mainly due to the large physical dimensions and/or large power requirements to realize autonomous navigation and obstacle detection.[13] For example, the simultaneous localization and mapping (SLAM) technique,[14] which is commonly used in robot navigation, relies on bulky and/or resource-consuming devices such as RGB-D (Red Green Blue-Depth) camera[15] or LiDAR (Light Detection and Ranging).[16] As a result, the proposed obstacle negotiation solutions[17,18] often require large and complex mechanical systems to avoid obstacles while traveling across uneven terrain, and the high power consumption of locomotion on mini robots, in the order of 10–1,000 mW, depletes their power source within minutes.[3–5]

Additionally, human detection is also another important criterion for USAR robots that most mini robots fail to achieve. As state-of-the-art human-detection methods are normally unsuitable for low-powered devices, there have been efforts to develop detection algorithms usable for low-power systems[2,19,20] that employ visible-light cameras, infrared (IR) cameras, or acoustic listening devices. Based on these works, mini robots with the ability to detect humans or objects were further explored, but their limitations such as low accuracy of sensors,[21] insufficient onboard computational resources,[22] and high power consumption by the locomotion system[23] have prevented their practical applications. Therefore, the creation of mini robots satisfying all the USAR requirements remains a challenge to be solved.

An alternative approach for these issues has emerged over the past decade in the form of insect–computer hybrid robots. These hybrid robots are the fusion of a living insect platform and a miniature electronic controller, created to combine the locomotion proficiency of an insect and the controllability of a robot. The earliest studies of insect control were demonstrated by Holzer et al. and Moore et al. in the late 20th century.[24,25] Latest development in insect–computer hybrid robots includes locomotion advancement such as free flight control of beetle with motion capture feedback[26,27] or walking control of beetle[28] and cockroach with integrated sound sensors.[29] Their low mass and volume, further enhanced by their intrinsic ability to walk, fly, and sense the surrounding environment, make them suitable candidates for the task of penetrating and navigating inside complex environments to search for a target.[29,30]

There are certain challenges that are unique to insect–computer hybrid robots that do not normally exist for artificial robots. These challenges include coarse navigation control due to the insects' voluntary action or habituation of electrical stimulus, and the limited environmental conditions under which the insect platforms can operate. Even so, the insect–computer hybrid robots can outperform their artificial counterpart by requiring much simpler hardware and control algorithm to operate in complicated unknown terrain, due to their natural ability to maneuver through obstacles.[31–33] In addition, as the insect platform is mainly controlled via electrical stimulus at neural, neuromuscular, or sensory sites,[34–36] their locomotion control can be achieved with power as few as 0.1 mW.[28,37] This leads to a much lower power consumption when compared with their artificial counterpart,[3–5] allowing them to participate in operations that last for more than an hour. The insects are also targeted as application platforms for latest advancements in fabrication technology such as miniature camera implementation on living insect.[38]

However, all recent studies of hybrid robots demonstrated operations of insect in an obstacle-free environment[39,40] or manually avoiding obstacle by navigating in a preplanned path,[41–43] despite having a natural obstacle negotiation ability that is beneficial for search-and-rescue missions. An onboard human-detection algorithm for such a system has not been thoroughly studied as well.[29,44,45] Existing literature suggested to use acoustic sensors for survivor detection.[29,44] For example, Eric Whitmire et al. developed a hybrid robot using omnidirectional microphones to detect help calls from survivors buried under rubbles.[44] However, this technique consumed a significant amount of power (i.e., 36 mW) and was computationally heavy, thereby preventing it from being implemented onboard.[29,44] To create an insect–computer hybrid robot applicable for search-and-rescue missions, the verification of a comprehensive hybrid robot for autonomous navigation through obstructed environments, human presence detection, and wireless communication must be addressed.

This study presents the first-ever demonstration of an insect–computer hybrid robot developed for USAR purposes that has the added capability of autonomous navigation in unknown environments with an external position tracking system, onboard human presence detection, and wireless communication, all integrated into a single hybrid robot (**Figure 1**a, Table S1, Supporting Information). A Madagascar hissing cockroach (*Gromphadorhina portentosa*) was used for the hybrid robot, which has a maximum payload limit of 15 g[29] and an average size of 6 × 2 cm. A custom backpack circuit board was mounted on the insect, which included the insect locomotion control system, navigation system, miniature IR camera, and wireless communication module (Figure 1b). The backpack was powered by a rechargeable LiPo battery (3.7 V, 120 mAh, 10 × 15 mm, 2.5 g, Figure 1b). The total weight of a fully assembled backpack, including the battery, was 5.5 g, which was safe within the insect's payload limit. A stimulation protocol and a simple and effective navigation program were implemented to utilize and harmonically cooperate with the insect's intrinsic obstacle negotiation ability in motion control (Figure 1c). In addition to that, onboard human detection was achieved by loading a lightweight yet accurate image classification model onboard that was based on histogram of oriented gradients (HOG) and support vector machine (SVM). The whole system was subsequently tested in an unknown environment to evaluate its performance (Movie S1, Supporting Information). The completed hybrid robot in this paper has a compact size and can navigate to predetermined destinations, autonomously traverse unknown obstacle–terrain, detect human presence, and wirelessly report the result to a remote console.

2. Results and Discussion

2.1. Navigation of Insect–Computer Hybrid Robot with Simple Feedback Control

The insect–computer hybrid robot consisted of an insect platform manipulated by a customized wireless backpack control

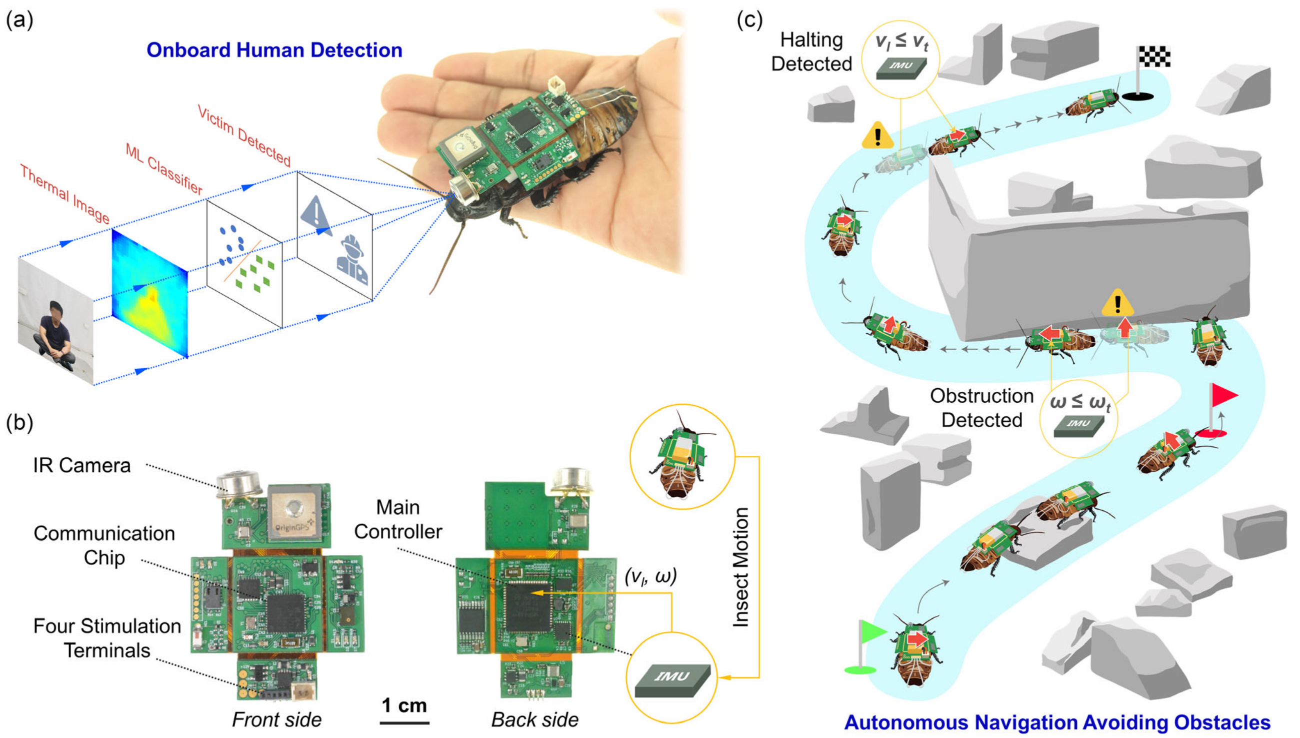

Figure 1. The insect–computer hybrid robot. a) The hybrid robot consists of a living Madagascar hissing cockroach and a wireless backpack controller. Autonomous navigation is enabled by electrically stimulating the insect's sensory system, while the infrared (IR) camera allows for onboard human detection via image classification. b) Main components of the backpack. The backpack is designed with necessary components to study the search-and-rescue activities of insect–computer hybrid robots. Wireless communication chip allows the communication with the rescue team base. IR camera provides information of survivors. A low-power microcontroller is used as the main control unit, which regulates the autonomous navigation as well as human-detection tasks. Onboard inertial measurement unit (IMU) provides the location of the insect for the navigation task. c) The established navigation algorithm cooperates harmoniously with the insect's natural locomotive ability to deal with unknown environments during the insect's exploration missions. By controlling the stimulation (denoted with red arrows), the algorithm directs the insect through obstacles to reach predetermined targets. Only data from the onboard IMU is used in navigation to predict and prevent potential navigation failure conditions (denoted with warning symbols).

system (Figure 1a,b). The insect's movement was directed by electrically stimulating its left and right cerci to induce turning; the insect made a right turn (clockwise rotation) when the left cercus was stimulated, and vice versa (Movie S2, Figure S1a, Supporting Information). This stimulation method did not hinder the insect's antennae, which act as crucial sensors for the insect to retain its original movement behavior.[31,46]

Based on this stimulation protocol, a simple on–off feedback control algorithm (**Figure** 2a), similar to those used in recent literature on navigation control of insect–computer hybrid robots,[39] was developed to evaluate the onboard automatic navigation capability of the insect in a predefined area (Figure 2b and S2, Supporting Information). This program used the position and orientation data of the insect provided in real time by a 3D motion capture system (Figure 2c) to steer the insect toward the destination. It was tested in low obstacle scenarios with 1.5 cm obstacle height[31] without prior knowledge of the environment (Figure 2b, $N = 5$ insects, $n = 49$ trials). The results (**Figure** 3a, Table S2, S3, Supporting Information) showed that the hybrid robot demonstrated superior capability of handling low obstacles, with only a simple feedback control navigation and no prior information of the obstacle.

The hybrid robot showed issues when it was tested with tall, wall-like obstacles at 10 cm height[47] ($N = 5$ insects, $n = 50$ trials). The simple navigation program cannot rely on insect obstacle negotiation behavior to overcome large path obstructions, thus it frequently failed to navigate the insect to the destination (Figure 3b, Table S2 and S3, Supporting Information). In 17 out of 37 failed navigation trials, the insect was immobilized while under electrical stimulation, in which the small θ (i.e., the angle made of the insect's orientation and the walls, Figure 2d) suggested that it was caused by physical obstruction from the tall obstacles (Figure 3c). To the contrary, 16 of 37 failed trials were caused by the insect's immobilization without electrical stimulation. In these cases, the navigation program deactivated the stimulation when the insect exhibited its obstacle negotiation behavior and reoriented itself to face toward the target, but the stimulation did not reactivate when the insect remained motionless (Figure 3d). In the last 4 out of 37 failed trials, the navigation process was negatively affected due to the contradiction between the insect's obstacle negotiation behavior and the navigation program. When the insect encountered a wall and retreated according to its natural behavior, it was trapped in that local vicinity when the navigation program stimulated it to move

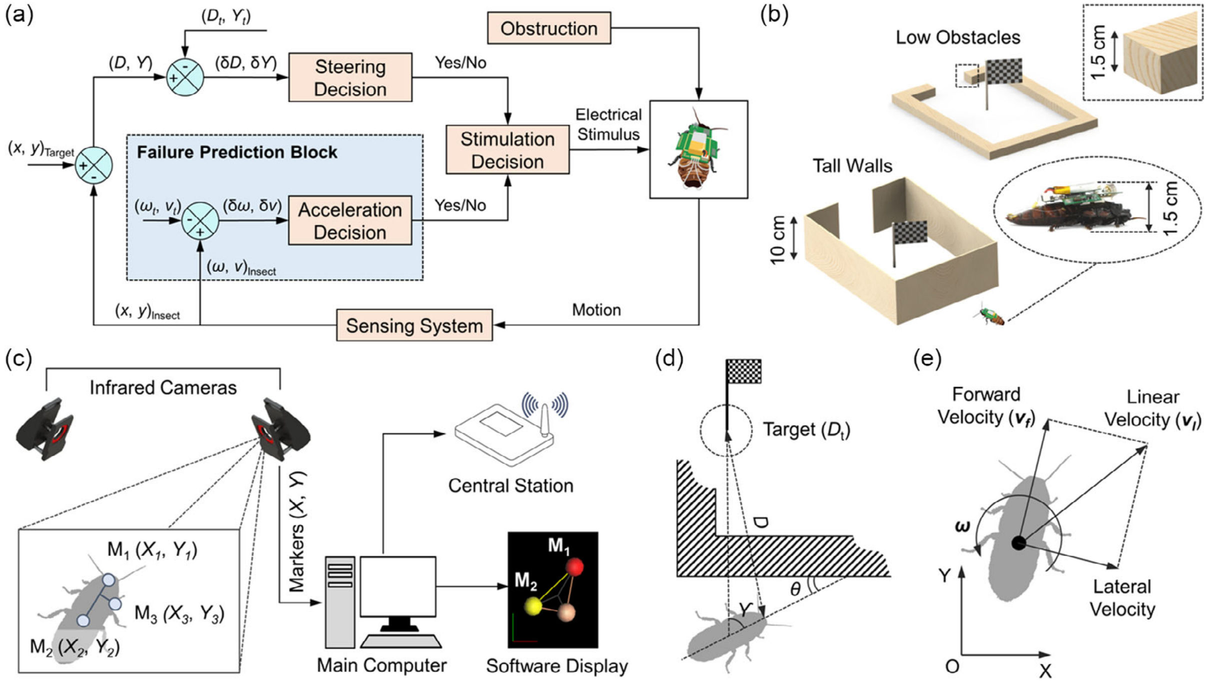

Figure 2. Control diagram of the "Predictive Feedback Control" navigation algorithm and the experimental setup. a) The Predictive Feedback Control algorithm is improved from the existing simple feedback control with the insertion of the "Failure Prediction Block." This block prevents the conflict between the insect's intrinsic behaviors and the navigation commands by monitoring the insect's motion (ω, v_l). If the monitored speeds fall below their thresholds (ω_t, v_t) indicating that the insect is halted by environment interaction, the steering stimulus is immediately discontinued and an acceleration stimulus is released to direct the insect out of its current vicinity, thereby preventing a navigation failure. b) Obstruction setup. The hybrid robot navigates across the arena with either low obstacles or tall walls to represent two different levels of obstruction complexity. The height of low obstacles is 1.5 cm, which is higher than the insect's usual height (≈1 cm). The height of tall walls is 10 cm. The characteristics of the two obstacles (e.g., shape, size, location) are unknown for the 3D motion capture system as well as the navigation program. c) The insect location is tracked using a 3D motion capture system. The system acquires the insect's anterior and center points via three retroreflective markers (i.e., M_1, M_2, M_3) mounted on a light carbon fiber frame (≈20 mg). The asymmetricity of the markers location allows the insect orientation to be recognized by the system. The marker coordinates are transferred to a computer, where they are displayed and logged for post-experimental analysis. The coordinates are sent wirelessly via a central station to the onboard navigation program as feedback data. d) Graphical description of navigation parameters. The angle Υ represents the angular displacement between the insect's orientation and the destination, calculated based on the insect's center point. The insect's anterior point is utilized to compute the distance D between itself and the destination. The insect reaches the target when D is smaller than its threshold D_t, which is the radius of the destination region. The acute angle made by the insect's body and the nearest obstacle is named as θ. e) The insect's velocities. Angular velocity (ω) is centered around the middle point of the insect. The linear velocity (v_l) is the superposition of two elements: the forward velocity (v_f, longitudinally aligned with the insect's orientation) and the lateral velocity (i.e., perpendicular with the insect's body).

toward the obstacle again, causing the whole process to be repeated (Figure 3e). This could be seen in all four trials, where nearly half of the total navigation time before timeout was spent on backward motion, that is, 40.93 ± 13.34 s out of 100 s (Figure 3f).

2.2. Enhanced Navigation Performance with Predictive Feedback Control

The three types of failures listed earlier showed the conflict between the insect's behavior and the navigation algorithm, especially when the algorithm failed to recognize and handle the immobility or "trapped state" (in the case of timeout failures) of the insect. This is because the simple feedback control only decides the turning direction based on the angular displacement between the insect's orientation and the destination but ignores the insect's natural movement. To overcome this shortcoming, a failure prediction algorithm was developed to improve the navigation performance in the tall obstacles situation, and subsequently implemented in a new navigation program, henceforth named as "Predictive Feedback Control" navigation (Figure 2a, S3, Supporting Information). The Predictive Feedback Control navigation improves upon the simple feedback control by involving insect's intrinsic natural obstacle avoidance ability. This allows the algorithm to heuristically predict whether the insect is blocked by obstacles. It is explained further in the next paragraph.

The Predictive Feedback Control navigation works by utilizing the fact that the insect typically walked straight without any external stimulus.[48,49] The navigation algorithm monitors linear speed of the insect (v_l, Figure 2e), then predicts and prevents immobility by prematurely accelerating it when v_l decrease below a given threshold (Table S4, Supporting Information). Similarly, a small measured ω (i.e., the insect's angular speed, Figure 2e) would indicate that the insect might be blocked by an obstacle, and it would otherwise cause the insect to either stay immobile or

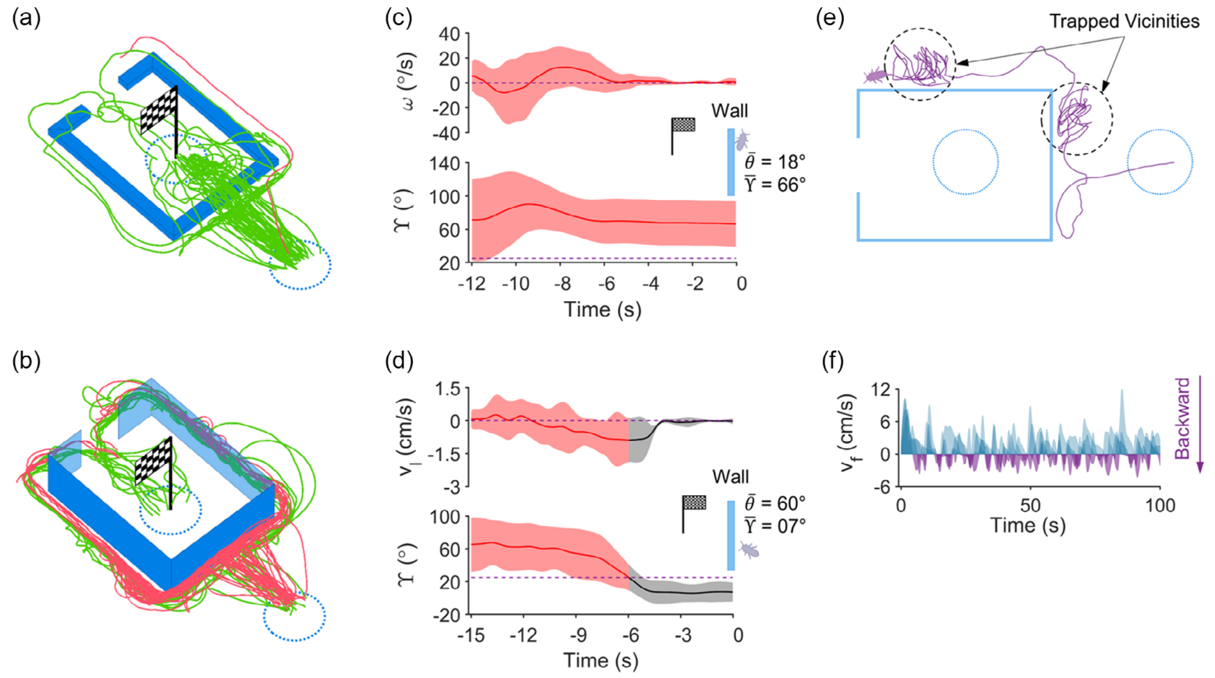

Figure 3. Contradiction of the existing simple feedback control navigation and the insect's behaviors. a,b) Trajectories of the insects in low obstacles and tall walls, respectively ($N = 5$ insects, $49 \leq n \leq 50$ trials for each configuration). Successful and failed navigations are plotted in green and red colors, respectively. In the cases of low obstacles, the simple algorithm attains a high success rate owing to the insect's ability to climb over the obstruction. However, the failure becomes dominant when the obstruction complexity is increased. This failure can be attributed to the conflict between the insect's behaviors and the navigation program. c,d) Anatomy of the first two types of failures: immobility with electrical stimulation and immobility without electrical stimulation. The stimulation and stimulation-free periods are denoted with red/black curves (mean) and shaded regions (standard deviation), respectively. The zeroth time mark indicates experiment termination. In (c), despite being stimulated (as $\Upsilon > \Upsilon_t$ – dash purple line), the induced angular speed of the insect reduces toward zero. Its angle θ is small (18°), implying that the insect is obstructed ($n = 17$ trials). In (d), the insect orients its body toward the destination (i.e., $\Upsilon < \Upsilon_t$) by moving backward (i.e., $v_f < 0$ cm s^{-1}) and then stops moving ($n = 16$ trials). e,f) Analysis of the third type of failure: timeout ($n = 4$ trials). The timeout is set at 100 s as a heuristically sufficient time for the insect to navigate to the destination (Navigation Experiment section). Figure (e) represents a typical case where the purple insect is trapped in some local vicinity as it continuously performs backward motion (i.e., $v_f < 0$ cm s^{-1}) to escape an obstacle but is then directed back to the obstacle, thus unable to reach the destination prior to timeout. The chart in Figure (f) plots the v_f against time in all 4 trials, with the blue regions depicting the forward motion (i.e., positive values) and the purple regions depicting the backward motions (i.e., negative values). The insect spends nearly half of the navigation time moving backward to reorient its body as an act of obstacle negotiation.

start a backward motion if not intervened. It is then possible to prevent both the immobility and timeout cases by interrupting the original stimulation and accelerating the insect away from the local vicinity when the insect's angular speed drops below the threshold (ω_t, Table S4, Supporting Information). The acceleration was attained by stimulating the insect's two cerci simultaneously (Movie S2, Supporting Information).[25]

In tall obstacle tests, the hybrid robot with Predictive Feedback Control navigation recorded a higher success rate of 94% (**Figure** 4a, $N = 5$ insects, $n = 49$ trials, Table S2 and S3, Supporting Information). As shown in a representative case (Figure 4d), during its stimulation states and free-walking states, the ω and v_l of the insect were respectively and periodically monitored by the navigation program (Figure 4e). In this example, there were four locations where ω and v_l were below their thresholds, each leading to an acceleration event. In the first and fourth events, the insect was walking freely while facing the destination when the v_l dropped below its given threshold. As a result, the navigation program accelerated the insect to keep it in motion, causing the insect to dash toward the obstacle and moved along the wall. This behavior was consistent with present studies on the insect's response to the obstacles situated on its escape path.[46,50] Such a reaction allowed the navigation algorithm to take advantage of the insect's natural obstacle negotiation reaction, which would be beneficial for cases where obstacle detection/crash prevention sensors could not be used. The second and third acceleration events occurred when ω was measured to be smaller than its threshold due to the insect being steered toward an obstacle. The stimulation was then interrupted by the navigation program, and the insect was accelerated so that it rapidly escaped the vicinity, preventing it from stopping or being trapped in the area.

For environments with low obstacles, this Predictive Feedback Control navigation retained a 100% success rate (Figure 4b, $N = 5$ insects, $n = 49$ trials, Table S2, S3, Supporting Information). There was a minor difference in navigation time, specifically 8.29 ± 6.29 s in simple feedback control and 6.35 ± 4.92 s in Predictive Feedback Control (Figure 4c, t-test, $p = 0.09$, $df = 95$). This can be attributed to the similar movement responses, clearly seen as the insect in both navigation programs displayed the major

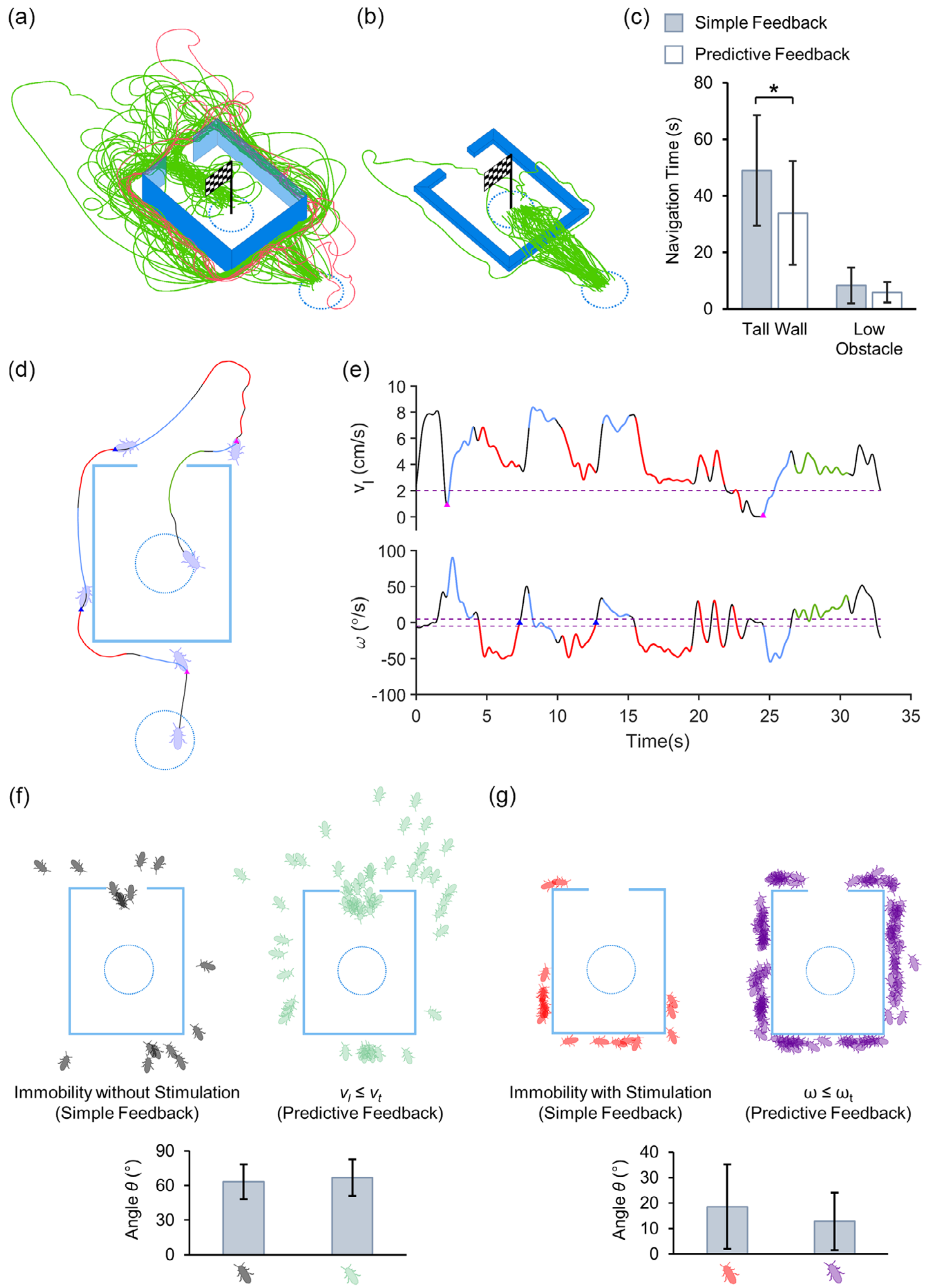

Figure 4. Performance of the Predictive Feedback Control algorithm. a,b) Navigation trajectories of the insect ($N = 5$ insects, $n = 49$ trials for each configuration). The navigation algorithm resolves the conflict between the insect and the navigation commands, resulting in the dominance of successful navigations (i.e., 45 green curves) over the failed ones (i.e., 4 red curves). In addition, the benefits of the insect's obstacle negotiation ability (i.e., climbing over low obstacles) are preserved. c) Comparison between the two navigation algorithms. Error bars represent 1-standard deviation. The asterisk indicates statistical significance. In the case of tall walls, the Predictive Feedback Control algorithm not only results in a higher success rate compared to its counterpart but also shortens the navigation time (t-test, $p < 0.01$, $df = 57$). Both algorithms perform similarly in the low obstacle case. d,e) Anatomy of a successful trial attained with the Predictive Feedback Control navigation. The black, green/red, and cyan segments each represent the free-walking period, left/right steering stimulation, and acceleration, respectively. The purple insect-like objects denote the insect's orientation, and the pink and blue triangles denote positions where v_l and ω fall below their thresholds (i.e., purple dashed lines), respectively. The accelerations fired by the Failure Prediction Block when the speeds fall below the threshold direct the insect out of failure-prone regions and kept it in motion. In addition, the insects avoid obstructions when dashing toward walls by changing their movement to follow the wall. f,g) Effect of the Predictive Feedback Control navigation in failure prevention. The figures display the comparison between the angle θ in motionless failures with and without stimulation in simple feedback control navigation and the angle θ recorded at locations where the corresponding speed fell below threshold in Predictive Feedback Control navigation. Error bars represent 1-standard deviation. Statistically, the similarity in the angle θ of both programs for each failure type implies that the new navigation program can foresee and prevent the potential failures that could occur with the simple feedback control navigation (t-test, $p = 0.80$, $df = 66$ for Figure (f), and $p = 0.07$, $df = 113$ for Figure (g)).

trend of climbing over low obstacles, that is, this behavior had a 96% rate of occurrence (Figure 3a and 4b).

Overall, the efficacy of the Predictive Feedback Control navigation algorithm to predict and prevent immobility of the insect could be observed by comparing the recorded angles θ for the two navigation programs in the same environment, in which there was a high correlation between the locations of acceleration events in the new navigation algorithm and the locations of the insect's immobility in the previous navigation algorithm (Figure 4f,g). The insect controlled by the new navigation algorithm also spent 5.15 ± 5.33 s in a backward motion, which was eight times lesser than the previous navigation program. With the reduction in time spent for backward motion, the success rate was improved by reducing the probability of timeout, and generally shortened the total navigation time from 48.98 ± 19.49 s to 33.91 ± 18.35 s (Figure 4c, *t*-test, $p < 0.01$, $df = 57$). These results showed that the new navigation algorithm successfully overcame the tall obstacle negotiation issue in hybrid robots. Nevertheless, it should be noted that there were three failed trials, that is, 6% out of all trials, recorded for the Predictive Feedback Control navigation, which was caused by the insect missing the obstacle entrance due to acceleration and thereby forcing the insect to navigate around the obstacles again until a timeout occurred (Figure 4a). However, this overshooting also occurred in other successful trials (Figure 4a), implying that these failed trials can be solved by optimizing the control parameters.

The new Predictive Feedback Control navigation algorithm is an important improvement compared to current navigation methods in insect–computer hybrid robots. The previous method navigated the insect at the expense of natural insect movement behavior; in this study, the Predictive Feedback Control navigation algorithm allows for the harmonic cooperation between the navigation program and insect's obstacle negotiation abilities, such that the insect can independently negotiate through obstacles encountered on its path while navigating toward the destination. This cooperation with the insect's natural movement behavior enables this simple navigation program to be exceptionally effective in handling unknown obstacles without any additional sensor requirements. A similar performance in artificial mini robots would require the inclusion of more obstacle recognition sensors that further increase its size and power requirements. In contrast, the hybrid robot used in this study reduces the computing and power consumed for the navigation process. This is important because it allows the robot to use these resources for other tasks, such as integrating an onboard human-detection system.

2.3. Onboard Human-Detection System

Considering the environment under rubble, an accurate human-detection system is required to achieve the goal of autonomous search and rescue in structural collapse disasters. The system must be able to operate under lightless conditions for at least several hours. An IR camera is an appropriate choice for such missions, as it can detect humans by capturing the thermographic difference between the human body and the surroundings.[51] However, due to high energy consumed by wireless communication, it is not possible to continuously stream data from the system to its control station. Therefore, all decisions regarding human presence detection must be made independently by the onboard system, subject to constraints on hardware size, power, and computational resource. The usage of an insect platform displays its benefit here by allowing much of the onboard resources to be saved for human detection.

The components, algorithm, and strategy for human detection have been carefully selected to achieve a fast, precise, and energy-efficient human-detection system that can be successfully implemented onboard. The human-detection system developed for the hybrid robot utilized an IR thermopile array with 32×32 pixels and $90° \times 90°$ field of view (Figure 1b and S1c, Supporting Information). This IR camera is suitable for insect mounting due to its low power consumption and small physical size. The main microcontroller denoises the IR images captured by the camera with a median filter[52] and then passes them to a machine-learning algorithm based on HOG and SVM that has been trained to recognize human heat signature[20] (**Figure 5**a). The human-detection results are then obtained and can be wirelessly sent to a remote base.

The machine-learning algorithm was built and tested on a computer to determine the suitable parameters to optimize the algorithm for onboard performance. A HOG feature descriptor with a cell size of 4×4 was selected after comparing the algorithm's accuracy with different cell sizes (Table S5, Supporting Information), and a linear SVM was used for the classifier due to its fast yet still sufficiently accurate classification results (Figure 5b,c). Next, a custom dataset (Table S6, Supporting Information) was prepared by capturing human images from the insect's point of view (POV) using the low-resolution 32×32 IR camera. Different human subjects and nonhuman hot objects were used for the training and testing dataset to prevent overfitting.

The established human-detection algorithm achieved 87% average accuracy in classifying between human and nonhuman subjects (Figure 5b), which is comparable to other studies on human classification using HOG and linear SVM.[20,53] Specifically, an average recall rate of 90% was achieved for human images captured within a distance of 0.5–1.5 m (Figure S4, Supporting Information). Additionally, the developed model occupied only 18.3 kB ($\approx$1%) of Flash and 52.2 kB ($\approx$20%) of Static Random Access Memory (SRAM) during onboard implementation (Table S7, Supporting Information), and its average computational time was only 95 ms. Although the hardware limitations caused the detection frame rate to be lower than similar pedestrian detection systems,[54] it was sufficient to capture up to 10 images per centimeter under the average insect speed of 3 cm s^{-1},[34] thus preserving the locality of each processed image and reducing the probability of overlooking the entrapped victims.

As long hours of operation are expected, the whole system was designed and developed to minimize total power consumption. The energy consumed by our human-detection algorithm running at 100% duty cycle was only 24 mW when embedded in the microcontroller. It consumed less energy than the human-detection technique used in a microphone-equipped backpack,[29,44] which detected sound sources and needed to consume 36 mW just for wireless audio transmission to a base station (personal computer (PC)) for remote processing. Similar human-detection algorithms were also implemented in

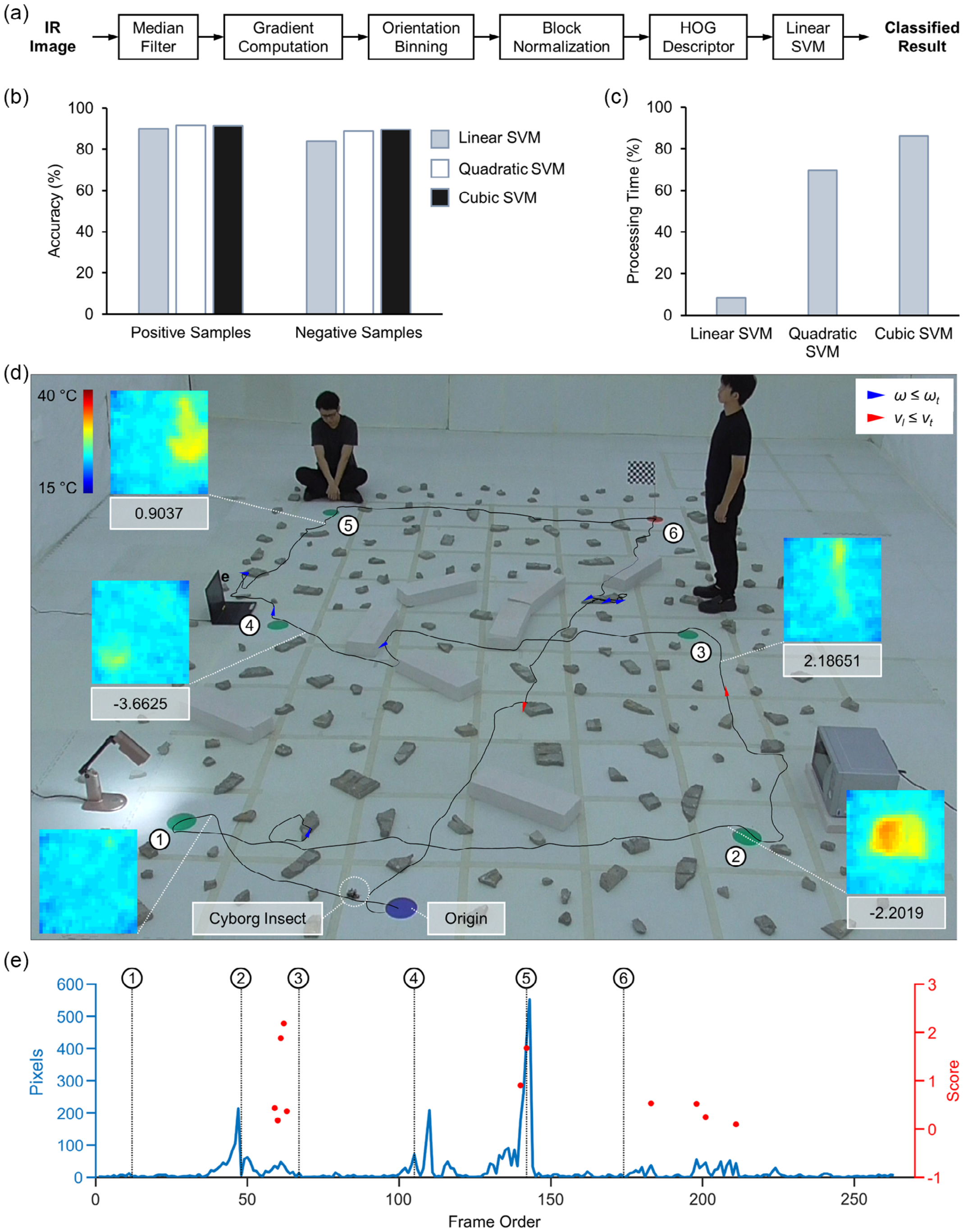

Figure 5. Overview of the human-detection algorithm and the demonstration in an unknown environment. a) The process of IR image classification to extract feature descriptors and determine the human presence in image. b) The accuracy and c) processing time are compared between three support vector machine (SVM) methods. While their detection accuracy is similar, linear SVM pulls ahead in terms of computational simplicity. d) In an environment with an unknown obstacle layout, the insect is able to reach all destinations (i.e., red/green 8 cm radius circles) in a predefined sequence (shown with the white numbered circles) before returning to the origin. The activations of Failure Prediction Block are denoted at instances where v_l or ω, respectively, fall below thresholds with red and blue arrows showing the insect's orientations. The onboard human-detection system operates at 1 Hz. Some captured IR images are shown with their corresponding scores and position where they are captured. The human-detection algorithm is only activated if the captured picture contains enough human-related information (i.e., when the number of pixels within the human temperature range is above its threshold of 15 pixels). As such, the left bottom-most picture does not possess a score. e) The plot depicts the operation of the human-detection algorithm during the demonstration. The pixel count that lies within the human temperature range is indicated by the blue curve. The output scores of possible human presences detected by the algorithm are denoted with red dots. The plot is temporally synced with the path in (d), with numbered circles referring to the same numbered locations in (d). The effectiveness of the onboard human-detection algorithm can be seen by referring to the corresponding subjects (i.e., either human or hot object), their location, and their respective scores along the horizontal axis.

field-programmable gate arrays (FPGAs)[54] and consumes 566 mW just to run the linear classification on a Virtex 7 platform.[20] In contrast, our insect–computer hybrid robot has a total system power consumption of 205.5 mW. Up to 2.2 h of operation time can be achieved with a 120 mAh battery, which is suitable for a practical USAR mission.

2.4. Demonstration of Insect–Computer Hybrid Robot Autonomously Searching for Human in an Unknown Environment

A demonstration of the insect–computer hybrid robot in a simplified search and rescue scenario was performed using the Predictive Feedback Control navigation program along with the onboard human-detection system. An insect–computer hybrid robot navigated autonomously through an uncharted terrain that included two humans and three hot objects with different shapes, sizes, and locations (Figure 5d, Movie S1, Supporting Information). A 3D motion capture system was used only to measure the position and orientation of the hybrid robot. The robot's linear speed and angular speed were calculated by an onboard inertial measurement unit (IMU).

As demonstrated, the Predictive Feedback Control navigation program succeeded in guiding the insect to reach all predetermined destinations in a set sequence, despite the lack of information regarding obstacles. The insect performed its obstacle negotiation ability by either climbing over the cement fragments or following the contour of obstacles, both with and without electrical stimulations (Figure 5d). There were seven times the insect was accelerated when the IMU measured its angular speed fell below the threshold, and two acceleration events were generated when the IMU measured a low linear speed. Simultaneously, the insect–computer hybrid robot actively searched for humans during the whole demonstration. The onboard human-detection system accurately distinguished between human and hot objects, shown in demonstration by correctly classifying the human 11 times while making zero misclassification mistakes on the nonhuman hot objects (Figure 5e). Overall, the results demonstrated the feasibility of the insect–computer hybrid robot in this simplified search and rescue mission where it must navigate into an unknown environment while scanning the area to locate victims.

3. Conclusion

We have presented an insect–computer hybrid robot with combined autonomous navigation and obstacle avoidance in an unknown environment and onboard human-detection system, which overcomes many flaws of USAR mini robots in terms of maneuverability and power consumption. The hybrid robot is capable of navigating and searching for victims in unknown environments and has a remarkably compact size yet exhibited a robust obstacle handling capability, owing to the custom navigation program that retains the natural locomotive ability of the insect. The inclusion of an onboard human-detection system using IR images removes the need for manual operation, allowing the hybrid robot to work autonomously and improve the overall searching efficiency. This study successfully proves the efficiency of a biological machine implementation in solving the issue of power–hungry actuators in artificial robots, thereby allowing the saved energy to be better utilized for other tasks, such as sensing and communication. There are improvements to be addressed in the hybrid robot design toward the implementation of insect–computer hybrid robots in real-life search-and-rescue missions. First, it is crucial for a localization system to be implemented onboard. A combination of IMU and UWB (Ultra-Wide Band) localization is being developed as a solution for this problem due to the power efficiency of IMU in providing position and orientation[55] and high penetration capability and high accuracy of UWB-positioning method.[56] The accuracy of such an onboard localization system should be at least 1.5 m, which can be lower than the 3D motion capture system. While high accuracy is not necessary as area-to-area navigation is preferred over point-to-point navigation in search-and-rescue missions, the navigation algorithm and searching strategy should be modified based on the accuracy. Second, it is essential to further improve the accuracy and operating range of the human-detection algorithm. This can be achieved by enhancing the sufficiency and quality of the thermal information provided to the algorithm using high-resolution IR cameras.[57,58] In addition, the algorithm's practicality can be strengthened by training it with more diverse datasets, such as those containing human images captured at different angles of view, which are likely to occur as the hybrid robots dynamically change their orientations when traversing complex terrains. Additionally, the backpack can be equipped with other sensors to enrich the collected information (e.g., CO_2 sensors[58]), thereby increasing reliability in survivor detection. In addition, the insect–computer hybrid robot has its own unique challenges to be addressed, for example, the coarse level of locomotion control leading to imperfect navigation path and difficulty in handling small opening in the rubble,[59] inability to operate in muddy or flooded terrain, and adverse effects caused by extreme environment conditions, such as in low-temperature situations. For the hybrid robot to smoothly crawl through obstacles, the backpack and the battery should be further miniaturized, which could be done by introducing energy harvesters and electric power generators such as thin-film solar cells and radiofrequency emitters.[60,61] Further investigation in expanding the roster of applicable insect platforms to fit with different working conditions, and swarm control of the hybrid robot to compensate the inaccuracy in locomotion control and improve coverage speed will be necessary to enable the participation of hybrid robots in real-life search-and-rescue missions.

4. Experimental Section

Animals: Male Madagascar hissing cockroaches (*Gromphadorhina portentosa*, 5.7 ± 0.6 cm, 6.32 ± 1.5 g) were kept in a laboratory terrarium (NexGen Mouse 500, Allentown Inc.) and were fed sliced carrots every week. The temperature and relative humidity were maintained at 25 °C and 60%, respectively. The use of the cockroach was permitted by the National Environmental Agency (Permit number NEA/PH/CLB/19-00012). Although ethical regulations for invertebrate research are still a subject of debate,[62] the study attempted to treat its experimental insects well and provide them with good living conditions. Specifically, the temperature, humidity, and feeding conditions were maintained appropriately for the insects by following a published guidance.[63] The insects were reared using a mouse housing system (NexGen Mouse 500, Allentown Inc.), which circulated clean air to the insects' territories, which were 19 × 13 × 38 cm plastic containers. These containers were washed weekly to maintain their hygiene. The number of insects in each container was limited to five to ensure that they had a spacious territory. They were fed weekly with various vegetables (e.g., carrots and apples). These conditions were applied not only to the intact insects (those to be used for the study) but also to the post-experimental insects (those were no longer being used for the study).

Electrode Implantation and Stimulation: Our method of steering the insect's locomotion was based on existing procedures.[25,64] The insect was first anaesthetized using carbon dioxide in an airtight container for 30 s. Sandpaper was used to file the mounting sites of the wireless stimulator and the implantation sites on the third abdominal segment, that is, the black dots located at the sides of the insect (Figure S1a, Supporting Information). This process allowed the backpack and beeswax to attach to the cuticle securely after implantation. Next, the cerci of the insect and their bases were covered with glue (Figure S1a, Supporting Information). After the glue solidified, the tips of the cerci were cut to create a small opening. A Teflon-insulated platinum wire (A–M Systems, coated diameter 0.14 mm, uncoated diameter 0.08 mm) was then inserted coaxially into the cerci, with an implantation depth of 10 mm from the tip of the electrode. The wire was de-insulated using tweezers prior to implantation such that the entire implanted portion was de-insulated. Beeswax was applied to secure the electrode in place and to seal the opening on the cercus.

For the third abdominal segment, an insect pin was used to create a small hole on each implantation site. A platinum wire was then inserted into the third segment while ensuring that the axis of the inserted wire was perpendicular to the surface of the cuticle. The implantation depth was 5 mm from the tip of the electrode. After implantation, beeswax was applied to cover the openings. The other ends of the implanted electrodes were connected to the wireless stimulator, which in turn would be mounted on the dorsal side of the insect throughout the experiments.

The insect was steered by passing a current through a pair of electrodes on one side, which induced the insect's rotation in the opposite direction (Movie S2, Supporting Information). The stimulation waveform was a biphasic square waveform of frequency 40 Hz and 50% duty cycle, in the range of 6–8 V.

Miniature Wireless Communication Stimulator (Backpack): The insect–computer hybrid robot was controlled by a customized wireless circuit board, or a backpack (Figure S1b,c, Supporting Information). Texas Instruments (TI) MSP432P4011 microcontroller (ARM 32-bit Cortex M4F, 48 MHz, 2 MB of Flash, 256 kB of SRAM) was used as the system's main controller unit for its excellent power efficiency. The stimulation signal was generated by the chip AD5504 (Analog Devices, Quad-channel, 12-bit, 7.3 mV resolution) with voltage supplied up to 12 V using TI TPS61046 Boost Converters. The backpack was integrated with an IMU using MPU9250 (InvenSense, 6 axes used with Digital Motion Processor, 11.55 mW) to detect the hybrid robot's movement. For human-detection purposes, an HTPA 32 × 32 IR camera was implemented for its small footprint and low power consumption (Heimann Sensor GmbH, 8 × 8 × 5 mm, 0.99 g, 21.45 mW average, 8 μW sleep, 32 × 32 resolution, 90° × 90° field of view). Additional sensors were embedded on the backpack to retrieve environmental information including temperature/humidity (BME280 by Bosch Sensortec) and volatile organic compounds (CCS811 by AMS). Backpack activities and data during the operation were recorded in the external onboard flash memory (MX25R1635F, 2 MB, Macronix) and were retrieved afterward. The insect–computer hybrid robot was controlled wirelessly via Bluetooth 5.1 (2.4 GHz) using CC1352 microcontroller unit (ARM 32-bit Cortex M4F, 48 MHz, 352 kB of Flash, 8 kB of SRAM). An attached 120 mAh LiPo battery (10 × 15 mm, 2.5 g) was used as a power source, increasing the total weight of the whole backpack to 5.5 g but was still safely within the maximum payload limit of the insect.[29]

Navigation Experiment: A batch of five insect–computer hybrid robots ($N = 5$ insects) was tested in two different terrains including low obstacles and tall walls (Figure S2, Supporting Information). Ten trials ($n = 10$ trials) for each hybrid robot were conducted on each combination of terrain and navigation algorithm, with a 5 min interval between each trial. The experiment for each terrain was performed on separate days. The same batch of insects was used for both navigation algorithms. The selection of two types of obstacles was to test the two navigation algorithms in the following scenarios: a) surmountable obstacles with heights similar to that of the insect and b) unsurmountable obstacles that were taller than the insect's height. The obstacle was selected as 1.5 cm in (a), according to past studies investigating the climbing motions of the insects.[31] In (b), the height of 10 cm was selected following existing studies on the insect's wall-following behaviors.[47]

Two 5 cm radius circles acted as the navigational origin and destination for the insect, which then navigated accordingly from one circle to another (Figure S2, Supporting Information). A successful trial was achieved if the insect reached the destination within 100 s. The trial was terminated for each of the following three conditions: when the insect reached the destination, when the insect was motionless for over 5 s, or when the experiment duration exceeded 100 s, whichever came first. The insect was defined to be motionless if its displacement was less than 0.5 cm.[65]

The success rate of each navigation algorithm was shown using the mean and 1-standard deviation of the success rates of all five experimented insects (Table S2, Supporting Information), which was defined to be the ratio of the number of successful trials to the number of total experimented trials.[39] Meanwhile, the navigation time of each navigation algorithm was represented using the mean and 1-standard deviation of the navigation time in successful trials, which was defined as the time taken for the insect to arrive at the destination. Analysis of variance test (0.05 significant level) was used to confirm that there was no significant effect of individual insects on this factor (Table S3, Supporting Information). Statistical *t*-test was used to compare the navigation time between the two algorithms, which was also employed to test the effectiveness of the "Failure Prediction Block" (Figure 4f,g).

The navigation experiment was conducted inside the viewable region of a 3D motion capture system (Vicon, Six T40 cameras, 4 Megapixel, 100 fps).[35,37] A structure made of three retroreflective markers and a carbon fiber frame was attached on the wireless backpack using double-sided tape to track the insect's location (Figure 2c).[28] Coordinates of these markers, which represent the insect's position and orientation, were provided by the 3D motion capture system to be used for deriving the motion of the insect during navigation. Specifically, the distance (D) and orientation ($\varUpsilon$) of the insect relative to the destination (Figure 2d) were employed in both navigation programs. They were calculated as

$$D = \sqrt{(X_1 - X_\mathrm{d})^2 + (Y_1 - Y_\mathrm{d})^2} \tag{1}$$

$$\varUpsilon = \cos^{-1} \frac{(X_1 - X_2)(X_\mathrm{d} - X_2) + (Y_1 - Y_2)(Y_\mathrm{d} - Y_2)}{\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}\sqrt{(X_\mathrm{d} - X_2)^2 + (Y_1 - Y_2)^2}} \tag{2}$$

where (X_d, Y_d), (X_1, Y_1), and (X_2, Y_2) were coordinates of the destination and two markers representing the insect's anterior and center points, respectively. The first was predetermined by users, while the latter two were provided by the 3D motion capture system.

A central station connected to the main PC was used to communicate with the backpack (Figure 2c). The coordinates of the three markers were streamed to the PC, which were then sent to the central station at the rate of 30 ms per package using a custom software written in C#. The location data was subsequently transferred wirelessly to the backpack so they could be used in the navigation feedback control algorithm.

The two navigation algorithms (simple feedback control and Predictive Feedback Control) were embedded into the backpack. These programs processed the location data and then issued the corresponding stimulation command to control the movement of the insect. The stimulation command was wirelessly transferred to the main PC via the central station for logging purposes as well as to synchronize with the location of the markers for post-experimental analysis.

Navigation Algorithms: In the simple feedback control navigation, the orientation of the insect relative to the destination, $\varUpsilon$ was computed every 30 ms. If $\varUpsilon$ exceeded its given threshold, $\varUpsilon_\mathrm{t}$ (°), a steering command was executed, otherwise, the insect could walk freely without any stimulation.

In the Predictive Feedback Control navigation (Figure S3, Supporting Information), a similar process occurred to decide whether the insect was 1) steered or 2) allowed to walk freely, but there were additional feedback processes for each state to predict the insect's stopping tendency and perform the required preventive measures. 1) When the insect was steered ($\Upsilon > \Upsilon_t$) and the duration of the steering stimulation exceeded d_s (ms), the insect's angular speed (ω) was measured at every predetermined interval t_v (ms). If ω was below its threshold ω_t (° s^{-1}), the navigation program judged the insect to be immobile and stopped the stimulation to allow free-walking state for t_{f1} (ms). The program then executed the acceleration command for a duration of d_a (ms). The insect was then allowed to walk freely for t_{f2} (ms) before the control loop returned to monitor Υ. 2) When the insect could walk freely ($\Upsilon \leq \Upsilon_t$), the program allowed the insect to maintain the free-walking state for t_{f3} (ms) if it was steered in the loop immediately preceding the current one. The linear speed (ν_l) was measured every t_v (ms) interval. If ν_l dropped below a given threshold ν_t (cm s^{-1}), the program executed the acceleration command for a duration of d_a (ms). The insect was then allowed to walk freely for t_{f2} (ms) before the control loop returned to monitor Υ.

The control parameters including Υ_t, t_v, t_{f1}, t_{f2}, t_{f3}, d_a, d_s, ω_t, ν_t were unchanged throughout the experiments (Table S4, Supporting Information).

In the navigation experiment, ω and ν_l were computed based on the location data of the three markers recorded by the 3D motion capture system. To mimic real deployment conditions, the onboard IMU was used instead for the demonstration. For the former case, an average-moving filter with the window size of 250 ms was used to compute ω and ν_l. For the latter case, ω and ν_l were computed using the data from the gyroscope and accelerometer. A low-pass filter of 10 Hz was implemented for noise removal.

Human-Detection Experiment: A machine-learning model for human detection was developed for this study. Since a low-powered processor was implemented, there were a few aspects to consider during the development. First, the memory must fit within the available space of the microcontroller, which was 1988 kB Flash and 191.8 kB Static RAM after deducting the memory required for other tasks (Figure S1b, Table S7, Supporting Information). Second, a fast-processing speed was required to achieve a real-time operation. Therefore, the complexity of the human-detection algorithm must be low to satisfy the two requests, yet its accuracy should be high to avoid missing potential human victims.

Images used in the training of the human-detection model were captured in our laboratory at normal room temperature (from 25 °C to 27 °C). The thermal images of two human subjects and several nonhuman hot objects varied in shapes and dimensions (Table S6, Supporting Information) were captured using the IR camera mounted on the backpack. Each subject was placed on a turning table (BLK ND-RC6013, 120 rpm, 60 cm diameter) during the data collection process to diversify its appearance in the image. The images for training dataset were captured at 0.5, 1.0, and 1.5 m to fit in the predefined range of the camera within 0.5–1.5 m, such that the characteristics of humans could be clearly seen in the captured image. In images containing human subjects, the number of pixels that fell into the human temperature range of 28–38 °C was found to be above 15 pixels. This number was then employed as a threshold to activate the onboard human-detection algorithm. The collected IR images were then preprocessed and denoised using a median filter (kernel size 3×3). After preprocessing, the final training dataset included 11 171 human images and 11 493 nonhuman images. These images were used to train the human-detection model.

The performance of the trained model was validated with a validation dataset, which was prepared with a similar procedure except for some changes made to increase the generality and variety of the images. Specifically, different human subjects and hot objects that have different shapes compared to that of the subjects used in the training data were captured (Table S6, Supporting Information), and the distance between the IR camera and the subjects was changed to increments of 0.1 m from 0.5 to 1.5 m.

For the machine-learning model itself, HOG was used as the feature descriptor to extract features from each thermal image. This method converted the distribution of the local intensity gradient and the orientation into a feature vector, which was used to characterize an object's shape and appearance in the image.[66] The feature vectors were passed to an SVM classifier which was commonly used in supervised machine-learning tasks.[20] Considering the 32×32 resolution of the IR images, a comparison was made between different cell sizes used for the descriptor including 2×2, 4×4, and 8×8, each of which divided the image into 225, 49, and 9 blocks containing 36 features, such that a feature vector length of 8100, 1764, and 324, respectively, was formed. In addition to that, three different SVM methods of Linear SVM, Cubic SVM, and Quadratic SVM were considered for the onboard human-detection classifier.

A comparison in accuracy and computational time between the three SVM methods was done to select the best performing model. Despite having a slightly lower accuracy compared to its nonlinear counterparts, linear SVM has the advantage of the lowest computational time owing to its simpler calculations.[54,66] Therefore, linear SVM was employed to study the accuracy of human detection under the three different HOG cell sizes and the cell size giving the highest accuracy would be selected for the HOG descriptor used in the final model. All models used in comparisons mentioned earlier were trained using MATLAB. The kernel parameters of the selected model were then implemented into the backpack's main microcontroller to realize the onboard human-detection system.

We used Code Composer Studio (CCS), an integraded development environment to build and transfer the human-detection program into the MSP432 microcontroller. To evaluate the impact of the human-detection algorithm on the whole system, CCS was used to monitor the hardware resources and measure the processing time for each captured image while running human detection. The EnergyTrace tool was used to measure the power, voltage, and current during circuit operation to calculate the power consumption.

Demonstration: To demonstrate the application in search-and-rescue missions, an insect–computer hybrid robot navigated autonomously through an unknown environment (Figure 5d, Movie S1, Supporting Information). Obstacles made of building materials (concrete and cement blocks) varying in shapes and sizes were situated randomly inside a 420×600 cm arena (Figure S5, Supporting Information). From the origin, the insect navigated through five predetermined targets made of 8 cm radius circles until it reached the destination, then it was directed back to the origin. The navigation was completed without sending the location data of the obstacles to the navigation program. The angular and linear speed measurements were done using an onboard IMU to simulate the actual use case of the insect–computer hybrid robot. The 3D motion capture system (Vicon) was used to obtain accurate insect location (X, Y, Z) to be fed to the onboard navigation program. Multiple hot objects and humans were introduced nearby the navigational targets to demonstrate the performance of the onboard human-detection system. The subjects were different from those used in the model training process. IR images captured by the IR camera were stored inside an external memory embedded on the backpack to be extracted later for analysis. The human detection results established from these images were wirelessly streamed to and displayed on the main PC.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

The authors thank Mr. Seah Hee Chuan and Dr. Goh Bing Hui, Terence at KLASS Engineering & Solutions Pte. Ltd, Mr. Cheng Wee Kiang, Mr. Ong Ka Hing, and Ms. Vanessia Choo at Home Team Science & Technology Agency (HTX), and distinguished officers at Singapore Civil Defence Force (SCDF) for their helpful comments and advice, Mr. Chew Hock See, Mr. Seet Thian Beng, Ms. Kerh Geok Hong Wendy, Mr. Roger Tan Kay

Chia, Mr. Li Rui, and Mr. Qifeng for their support. This work was supported by KLASS Engineering & Solutions Pte. Ltd (NTU REF 2019-1585).

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

H.S., P.T.T.N., H.D.N., F.C., T.T.V.D., and T.L.N. conceived and designed the research; H.D.N., P.T.T.N., F.C., K.K., and J.H.G. established insect control protocol; T.L.N., V.T.D., D.L.L., H.D.N., B.S.C., P.T.T.N., and Y.L. developed hardware and software for the backpack; P.T.T.N., B.S.C., and H.D.N. developed the human-detection algorithm; P.T.T.N., H.D.N., D.L.L., and B.S.C. conducted the experiment and analysis; P.T.T.N., H.D.N., D.L.L., B.S.C., and H.S. wrote and edited the manuscript; H.S. supervised the research. All authors read and edited the paper.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

autonomous navigation, cyborg insect, human detection, obstacle negotiation, search-and-rescue, thermal image, unknown environment

Received: September 16, 2022
Revised: December 21, 2022
Published online: January 27, 2023

[1] J. L. Casper, M. Micire, R. R. Murphy, in *Unmanned ground vehicle technology II*, Orlando, FL, United States **2000**, pp. 292–302.

[2] J. Wong, C. Robinson, *Urban Search and Rescue Technology Needs: Identification of Needs*, Federal Emergency Management Agency (FEMA) and the National Institute of Justice (NIJ), USA **2004**.

[3] P. Birkmeyer, K. Peterson, R. S. Fearing, in *2009 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, IEEE, Piscataway, NJ **2009**, pp. 2683–2689.

[4] A. M. Hoover, E. Steltz, R. S. Fearing, in *2008 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, IEEE, Piscataway, NJ **2008**, pp. 26–33.

[5] B. Goldberg, R. Zufferey, N. Doshi, E. F. Helbling, G. Whittredge, M. Kovac, R. J. Wood, *IEEE Rob. Autom. Lett.* **2018**, *3*, 987.

[6] Y. Chen, N. Doshi, R. J. Wood, *IEEE Rob. Autom. Lett.* **2020**, *5*, 4820.

[7] N. T. Jafferis, E. F. Helbling, M. Karpelson, R. J. Wood, *Nature* **2019**, *570*, 491.

[8] M. J. Anderson, J. G. Sullivan, J. L. Talley, K. M. Brink, S. B. Fuller, T. L. Daniel, in *2019 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, NJ **2019**, pp. 6077–6082.

[9] Y. Wu, J. K. Yim, J. Liang, Z. Shao, M. Qi, J. Zhong, Z. Luo, X. Yan, M. Zhang, X. Wang, R. S. Fearing, *Sci. Rob.* **2019**, *4*, eaax1594.

[10] J. Liang, Y. Wu, J. K. Yim, H. Chen, Z. Miao, H. Liu, D. Wang, W. Qiu, Z. Shao, *Sci. Rob.* **2021**, *6*, eabe7906.

[11] M. J. Anderson, J. G. Sullivan, T. K. Horiuchi, S. B. Fuller, T. L. Daniel, *Bioinspiration Biomimetics* **2020**, *16*, 026002.

[12] V. Iyer, H. Gaensbauer, T. L. Daniel, S. Gollakota, *Nature* **2022**, *603*, 427.

[13] R. S. Pierre, S. Bergbreiter, *Ann. Rev. Control Rob. Auton. Syst.* **2019**, *2*, 231.

[14] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, *IEEE Trans. Rob.* **2016**, *32*, 1309.

[15] C. Wang, L. Meng, S. She, I. M. Mitchell, T. Li, F. Tung, W. Wan, M. Q. H. Meng, C. W. de Silva, in *2017 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS))*, IEEE, Piscataway, NJ **2017**, pp. 109–116.

[16] K. Qian, X. Ma, F. Fang, X. Dai, B. Zhou, *Int. J. Adv. Rob. Syst.* **2016**, *13*, 1729881416670902.

[17] D. Goldschmidt, F. Wörgötter, P. Manoonpong, *Front. Neurorob.* **2014**, *8*, 3.

[18] J. Kim, J. Kim, D. Lee, *J. Mech. Sci. Technol.* **2018**, *32*, 5389.

[19] S. Jagannathan, K. Desappan, P. Swami, M. Mathew, S. Nagori, K. Chitnis, Y. Marathe, D. Poddar, S. Narayanan, A. Jain, in *2017 IEEE Int. Conf. Consumer Electronics (ICCE)*, IEEE, Piscataway, NJ **2017**, pp. 233–234.

[20] H. Madadum, Y. Becerikli, in *2017 10th Int. Conf. Electrical and Electronics Engineering (ELECO)*, IEEE, Piscataway, NJ **2017**, pp. 1286–1290.

[21] A. Himoto, H. Aoyama, O. Fuchiwaki, D. Misaki, T. Sumrall, in *IEEE Int. Conf. Mechatronics, 2005. ICM'05*, IEEE, Piscataway, NJ **2005**, pp. 526–531.

[22] S. Balasubramanian, Y. M. Chukewad, J. M. James, G. L. Barrows, S. B. Fuller, in *2018 7th IEEE Int. Conf. Biomedical Robotics and Biomechatronics (Biorob)*, IEEE, Piscataway, NJ **2018**, pp. 1297–1302.

[23] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, L. Benini, *IEEE Internet Things J.* **2019**, *6*, 8357.

[24] R. Holzer, I. Shimoyama, in *Proc. 1997 IEEE/RSJ Int. Conf. Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, Vol. *3*, IEEE, Piscataway, NJ **1997**, pp. 1514–1519.

[25] T. E. Moore, *Acta Entomol. Slov.* **1998**, *6*, 71.

[26] Y. Li, H. Sato, B. Li, *IEEE Trans. Rob.* **2021**, *37*, 2041.

[27] T. T. Vo-Doan, V. T. Dung, H. Sato, *Cyborg Bionic Syst.* **2022**, *2022*, 1.

[28] H. D. Nguyen, P. Z. Tan, H. Sato, T. T. Vo-Doan, *IEEE Trans. Med. Rob. Bionics* **2020**, *2*, 331.

[29] T. Latif, E. Whitmire, T. Novak, A. Bozkurt, *IEEE Sens. J.* **2016**, *16*, 3444.

[30] H. Sato, M. M. Maharbiz, *Front. Neurosci.* **2010**, *4*, 199.

[31] C. Harley, B. English, R. Ritzmann, *J. Exp. Biol.* **2009**, *212*, 1463.

[32] B. Blaesing, H. Cruse, *J. Exp. Biol.* **2004**, *207*, 1273.

[33] S. Ma, P. Liu, S. Liu, Y. Li, B. Li, *IEEE Trans. Rob.* **2022**, *38*, 2208.

[34] T. Latif, A. Bozkurt, in *2012 Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, IEEE, Piscataway, NJ **2012**, pp. 972–975.

[35] F. Cao, C. Zhang, H. Y. Choo, H. Sato, *J. R. Soc. Interface* **2016**, *13*, 20160060.

[36] C. Sanchez, C.-W. Chiu, Y. Zhou, J. González, S. B. Vinson, H. Liang, *J. R. Soc. Interface* **2015**, *12*, 20141363.

[37] F. Cao, C. Zhang, T. T. Vo Doan, Y. Li, D. H. Sangi, J. S. Koh, N. A. Huynh, M. F. B. Aziz, H. Y. Choo, K. Ikeda, P. Abbeel, *Plos One* **2014**, *9*, e105389.

[38] V. Iyer, A. Najafi, J. James, S. Fuller, S. Gollakota, *Sci. Rob.* **2020**, *5*, eabb0839.

[39] E. Whitmire, T. Latif, A. Bozkurt, in *2013 35th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, Japan **2013**, pp. 1470–1473.

[40] H. D. Nguyen, V. T. Dung, H. Sato, T. T. Vo-Doan, *Sens. Actuators, B* **2023**, *376*, 132988.

[41] T. Latif, E. Whitmire, T. Novak, A. Bozkurt, in *2014 36th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, IEEE, Piscataway, NJ **2014**, pp. 1670–1673.

[42] M. Ariyanto, C. M. M. Refat, X. Zheng, K. Hirao, Y. Wang, K. Morishima, *Computation* **2022**, *10*, 179.

[43] Y. Tsukuda, D. Tagami, M. Sadasue, S. Suzuki, J.-L. Lu, Y. Ochiai, in *CHI Conf. Human Factors in Computing Systems Extended Abstracts*, New Orleans, LA, USA **2022**, pp. 1–13.

[44] E. Whitmire, T. Latif, A. Bozkurt, in *SENSORS, 2014 IEEE*, IEEE, Piscataway, NJ **2014**, pp. 2195–2198.

[45] A. Bozkurt, E. Lobaton, M. Sichitiu, T. Hedrick, T. Latif, A. Dirafzoon, E. Whitmire, A. Verderber, J. Marin, H. Xiong, in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIII*, Baltimore, Maryland, United States **2014**, pp. 498–503.

[46] Y. Baba, A. Tsukada, C. M. Comer, *J. Exp. Biol.* **2010**, *213*, 2294.

[47] J.-M. Mongeau, A. Demir, J. Lee, N. J. Cowan, R. J. Full, *J. Exp. Biol.* **2013**, *216*, 4530.

[48] R. Jeanson, S. Blanco, R. Fournier, J.-L. Deneubourg, V. Fourcassié, G. Theraulaz, *J. Theor. Biol.* **2003**, *225*, 443.

[49] K. A. Daltorio, B. T. Mirletz, A. Sterenstein, J. C. Cheng, A. Watson, M. Kesavan, J. A. Bender, R. E. Ritzmann, R. D. Quinn, in *Conf. Biomimetic and Biohybrid Systems*, Cham **2014**, pp. 72–83.

[50] J. M. Camhi, E. N. Johnson, *J. Exp. Biol.* **1999**, *202*, 631.

[51] E. Levin, J. L. McCarty, J. Bialas, A. Banaszek, S. Banaszek, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2016**, *8*, 99.

[52] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Pearson/ Prentice Hall, Hoboken, New Jersey, U.S **2008**.

[53] X. Cao, C. Wu, P. Yan, X. Li, in *2011 18th IEEE Int. Conf. Image Processing*, IEEE, Piscataway, NJ **2011**, pp. 2421–2424.

[54] N. Attarmoghaddam, K. F. Li, in *2019 IEEE Pacific Rim Conf. Communications, Computers and Signal Processing (PACRIM)*, IEEE, Piscataway, NJ **2019**, pp. 1–7.

[55] J. Cole, A. Bozkurt, E. Lobaton, *Sensors* **2020**, *20*, 4486.

[56] A. Poulose, O. S. Eyobu, M. Kim, D. S. Han, in *2019 Eleventh Int. Conf. Ubiquitous and Future Networks (ICUFN)*, Zagreb, Croatia **2019**, pp. 84–88.

[57] B. Triggs, S. Ito, K. Yokoi, *Human Detection in Thermal Images Using Low-Level Features*, PAK Publishing House, Poland **2015**.

[58] D. Zhang, S. Sessa, R. Kasai, S. Cosentino, C. Giacomo, Y. Mochida, H. Yamada, M. Guarnieri, A. Takanishi, *Sensors* **2018**, *18*, 852.

[59] E. Ackerman, in *IEEE Spectrum*, ed, IEEE, Piscataway, NJ **2021**.

[60] H. Jinno, K. Fukuda, X. Xu, S. Park, Y. Suzuki, M. Koizumi, T. Yokota, I. Osaka, K. Takimiya, T. Someya, *Nat. Energy* **2017**, *2*, 780.

[61] T. Ozaki, N. Ohta, T. Jimbo, K. Hamaguchi, *Nat. Electron.* **2021**, *4*, 845.

[62] C. B. Freelance, *Sci. Eng. Ethics* **2019**, *25*, 1339.

[63] P. Mulder, A. Shufran, *Madagascar Hissing Cockroaches: Information and Care*, Oklahoma Cooperative Extension Service, Oklahoma State, USA **2016**.

[64] J. C. Erickson, M. Herrera, M. Bustamante, A. Shingiro, T. Bowen, *Plos One* **2015**, *10*, e0134348.

[65] A. Dirafzoon, T. Latif, F. Gong, M. Sichitiu, A. Bozkurt, E. Lobaton, in *2017 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Piscataway, NJ **2017**, pp. 2457–2461.

[66] N. Dalal and B. Triggs, in *2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05)*, vol. *1*, IEEE, Piscataway, NJ **2005**, pp. 886–893.