Short *et al*. *Intell Robot* 2023;3(2):222-41
DOI: 10.20517/ir.2023.14

Intelligence & Robotics

Research Article

Open Access

A bio-inspired algorithm in image-based path planning and localization using visual features and maps

Daniel Short[1,#], Tingjun Lei[1,#], Chaomin Luo[1], Daniel W. Carruth[2], Zhuming Bi[3]

[1]Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA.
[2]Center for Advanced Vehicular Systems, Mississippi State University, Mississippi State, MS 39762, USA.
[3]Department of Civil and Mechanical Engineering, Purdue University Fort Wayne, Fort Wayne, IN 46805, USA.
[#]The authors contributed equally.

Correspondence to: Prof. Chaomin Luo, Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA. E-mail: Chaomin.Luo@ece.msstate.edu

How to cite this article: Short D, Lei T, Luo C, Carruth DW, Bi Z. A bio-inspired algorithm in image-based path planning and local-ization using visual features and maps. *Intell Robot* 2023;3(2):222-41. http://dx.doi.org/10.20517/ir.2023.14

Received: 19 Feb 2023 **First Decision:** 13 Apr 2023 **Revised:** 26 May 2023 **Accepted:** 5 Jun 2023 **Published:** 28 Jun 2023

Academic Editors: Simon X. Yang, Jiankun Wang **Copy Editor:** Yanbing Bai **Production Editor:** Yanbing Bai

Abstract

With the growing applications of autonomous robots and vehicles in unknown environments, studies on image-based localization and navigation have attracted a great deal of attention. This study is significantly motivated by the observation that relatively little research has been published on the integration of cutting-edge path planning algorithms for robust, reliable, and effective image-based navigation. To address this gap, a *biologically inspired* Bat Algorithm (BA) is introduced and adopted for image-based path planning in this paper. The proposed algorithm utilizes visual features as the reference in generating a path for an autonomous vehicle, and these features are extracted from the obtained images by convolutional neural networks (CNNs). The paper proceeds as follows: first, the requirements for image-based localization and navigation are described. Second, the principles of the BA are explained in order to expound on the justifications for its successful incorporation in image-based navigation. Third, in the proposed image-based navigation system, the BA is developed and implemented as a path planning tool for global path planning. Finally, the performance of the BA is analyzed and verified through simulation and comparison studies to demonstrate its effectiveness.

Keywords: Image-based navigation, Bat Algorithm, path planning, localization, autonomous vehicles, Mapping

1. INTRODUCTION

Autonomous vehicle path planning plays a crucial role in Simultaneous Localization and Mapping algorithms (SLAM) and navigation modules. Its purpose is to construct a safe and collision-free trajectory, including map-based path planning and image-based path planning[1–9]. Map-based navigation utilizes sensor readings performed in Cartesian space with terrain costs in maps[10–15]. Image-space of an on-board camera is used to feed raw images into a cost-image to form image-based path planning[16–18]. The constructed maps have various fundamental representations, such as metric maps, geometric maps, feature maps, and hybrid models. These representations essentially are an integration of multiple methods. As the outdoor landscape tends to be myopic, the limited range of LIDAR sensors poses challenges in building up maps with long-distance perception. Consequently, this limitation results in inefficient path planning in sensor-based navigation systems[19–22]. In map-based path planning, obstacles can often present aliasing issues when converting from sensor readings to the Cartesian space. For instance, in sensor-based navigation, solid areas may have gaps or slender corridors may disappear completely. However, when we consider an image feature-based navigation, researchers can represent such an image-based map by defining a graph, with the set of vertices representing images and the set of edges representing paired relationships between these images. Maps may be constructed in light of the sequence of images acquired to find paths from the initial to the destination pose. Image-based localization, with feature representations for image retrieval and image processing, contributes to mapping and navigation.

1.1. Related work

Navigation in an unknown and unstructured environment is extremely challenging and is involved in multiple, well-known, and well-studied disciplines[3,23–26]. Within the research community, there is a body of knowledge relating to neural network (NN) architectures that apply machine learning concepts to extract features and image descriptors from scenes, thereby facilitating various tasks related to autonomous vehicles[27,28]. This is typically accomplished through the implementation of a pooling layer inspired by the Vector of Locally Aggregated Descriptors (VLAD), which pools extracted descriptors into a fixed image representation, allowing its NN parameters to be learned by back-propagation NN. VLAD descriptors accomplish visual location recognition by directly using the intermediate layer outputs of pre-trained CNNs[29].

Visual place recognition methods can be based on directed acyclic graph matching, where feature maps extracted from deep CNNs are used to form probability distributions on image representation[30]. Another approach is based on extreme K-means and effective extreme learning machines (EELMs), where image decomposition with curvelet transformation is used to reduce dimensionality and generates a set of distinctive features[31] for use in future processing phases. As an example, Muthusamy *et al.*[32] applied a Gaussian mixture model (GMM) in various pattern recognition applications with EELM with excellent results.

The developments in feature extraction through NNs and CNNs naturally lend themselves to the performance of tasks associated with autonomous vehicles, such as SLAM[33]. One particular type of SLAM, Learning SLAM, can obtain camera pose and create a 3D map. However, it needs to have the prior dataset to train the network, and its performance often depends on that dataset. As a result, learning SLAM has poor generalization and is less versatile than some of its other implementations, such as geometric SLAM. This is generally owing to its 3D map being less accurate than one created via geometric SLAM. Visual SLAM, on the other hand, is a more dynamic system that typically consists of three components: (1) a visual odometry algorithm that provides an initial state estimate; (2) a place recognition system that is able to relate the currently observed scene to previously observed scenes; and (3) an optimization back-end which consistently integrates the place matches from the place recognition system with the full stream of state estimates[34]. While visual SLAM has proven to be highly valuable in ideal environments, there are certain disadvantages to this methodology that researchers are actively addressing in current research. The fundamental issue lies in the challenges posed by environmental changes, which induce errors and make feature identification difficult. These challenges can

occur due to changes in time, environment, camera posture, and other factors[35].

Various forms of deep NNs (DNNs) and artificial intelligence (AI) techniques are utilized to make a system more robust by autonomously tuning parameters, thereby enhancing overall performance. Learned features, such as those derived from massive amounts of deep learning (DL) meta-sensor data, are aggregated into coherent large-scale maps and then classified features extracted from those DNNs, such as "lake", "road", "field", "obstacles", or "traversable terrain", assigned with corresponding cost values in a Cartesian space[36]. Graphs are then constructed based on these maps to enable path planning. Training data can be constructed either directly from image-space features or by projecting pixel data into Cartesian space, depending on the methods employed. To create an optimal trajectory, DL-based methods are used to construct a path that takes into account the geometry of the explored areas, taking into account the starting and ending positions. Outputs of image processing techniques, such as edge detection and region segmentation, are used to determine the explorable regions of the map[17]. Path planning inputs are derived from standard commercially available NNs, which are used to generate topological maps. Image-sensor input data are represented as a bipartite cost-graph, with disjoint nodes representing image-feature distances and their corresponding geographic distances[37].

In a parallel research area, biologically inspired algorithms have become popular for performing optimizations and solving problems involving n-dimensional search spaces. Evolutionary Algorithms (EAs) are widely used as global searching methods for optimization, and hybrids of EAs and analytical methods are providing a promising atmosphere for NN and CNN training applications[38]. Ongoing research in this domain provides fuel for the integration of subsets of EA-like swarm methodologies for parameter estimation and optimization. The swarm intelligence methodology aims to optimize the conditioning of EELM, and some researchers have already proposed an effective particle swarm optimization (PSO) algorithm known as the Multitask Beetle Antennae Swarm (MBAS) Algorithm. This algorithm was inspired by the structures of the artificial bee colony (ABS) algorithm and the Beetle Antennae Search (BAS) algorithm[39]. In fact, some research efforts even claim that the approach is so straightforward and effective that it may be possible to completely replace traditional NN training paradigms, including back-propagation[40].

EELM is not the only area benefiting from swarm methodologies such as PSO. The controller of autonomous vehicles has seen significant improvements as well, particularly in the case of four-wheel steer four-wheel drive (4WS4WD) vehicles. The improved PSO-based controller takes into account all the slip forces acting on the vehicle, leading to a notable enhancement in its robustness[41–43]. Population-based algorithms have been used in a more generalized way to solve optimization problems that are typically challenging for traditional deterministic algorithms, such as the issue of the local optima. One novel population based on an optimization strategy involves leveraging the different behaviors of mosquitoes during the foraging process[44]. Another biomimetic swarm intelligence optimization algorithm, pigeon-inspired optimization (PIO), has also been introduced as a novel approach in this field. PIO simulates the homing behavior of pigeons using magnetic fields and landmarks. PIO has also been further improved to enhance its effectiveness in solving many-objective optimization problems (MaOPs)[45]. The list of EA research is extensive. Some of the optimization algorithms currently published include bacterial colony chemotaxis (BCC) optimization[46], the "swarm of bees" or BSO algorithm[47], and the discrete artificial bee colony (DABC) algorithm[48].

One important bio-inspired optimization algorithm is the Bat Algorithm (BA). The BA is the focus of our current research and has been the subject of many research papers. Various variations to the BA have been proposed, demonstrating its versatility and effectiveness in solving a wide range of optimization problems across multiple fields. This study showed that the proposed BA algorithm clearly outperformed competing methods, such as the Pareto concept, in terms of performance and solution quality. Furthermore, due to the similarities between the BA and the PSO, BA has also been applied to solve the Wireless Sensor Network (WSN) coverage problem. In the context of wireless sensor node development research, the sensor node itself is regarded as a

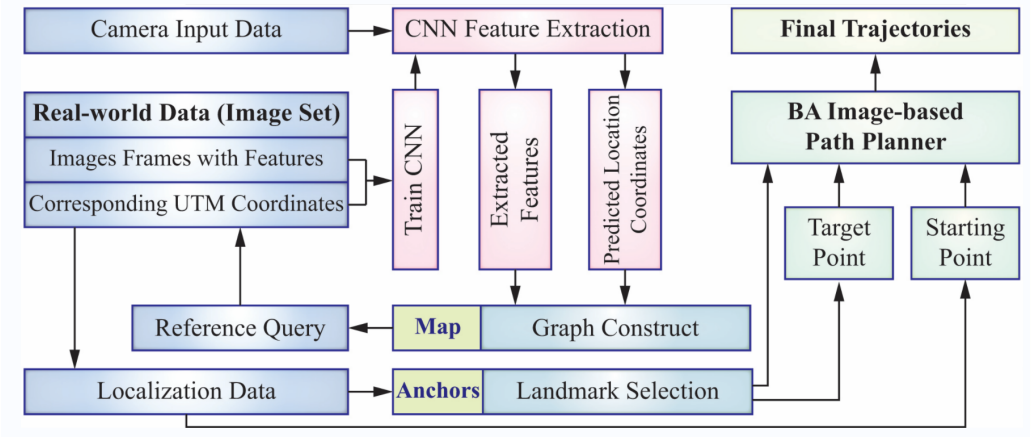

Figure 1. Proposed framework for Bat Algorithm in image-based path planning and localization using visual features and maps.

bat, and its position is regarded as the position of the bat. By implementing the BA, significant improvements in the accuracy of node positioning in practical and complicated environments have been demonstrated[49]. Additionally, an improved version of the BA combined with Differential Evolution (DE), known as IBA, has been proposed to optimize the three-dimensional path planning problem for Uninhabited Combat Air Vehicles (UCAV)[50]. This application of BA in UCAV path planning is notable because it was a novel approach at the time. The knowledge of this fact serves as the primary motivator for our current research. If BA can be used to plan and optimize 3D paths in combat aircraft, it is reasonable to believe that it could yield similar benefits for autonomous ground vehicles[51].

In the related work reviewed, *it is found* that no **BA** (or any other biologically inspired optimization) has been applied to the SLAM path planning problem. Since SLAM and seqSLAM utilize visual features and CNNs to produce a global path through a sequence of images, the implementation of a BA would be a novel and useful application of the optimization capabilities for optimizing the shortest path. This would involve computing a local path once the global trajectory has already been computed.

1.2. Proposed framework and original contributions

Studies on image-based navigation have become increasingly popular due to the increasing interest in their numerous applications in the field of autonomous vehicles[52,53]. In particular, several frameworks have been proposed and prototyped for image-based navigation. For example, Thoma *et al.*[37] developed a framework for image-based mapping, localization, and navigation of autonomous vehicles, and the algorithms for map constructions and self-localizations were discussed in detail. However, the algorithms required for path planning play an irreplaceable role in making an integrated imaged-based navigation system successful. Although limited work has been done on integrating path planning algorithms into learning systems, it has been noted that a reliable, biologically inspired, or nature-inspired optimization algorithm is missing from the image-feature-based navigation path planning models[54]. Therefore, this paper aims to develop a more robust path planner and evaluate the improved navigation results in comparison with Thoma *et al.*[37]. The BA is proposed and integrated as the state-of-the-art path planner into the program flow, and it is verified by measuring the performance of the outcomes based on the same datasets in the reference[55]. The flowchart of the proposed framework is presented in Figure 1.

The main contributions of this paper are summarized as follows:

- A biologically inspired image-based path planning and localization framework is proposed for robust, reliable, and effective image-based navigation.
- An image-based navigation is developed to use visual features and mapping that satisfy the require-

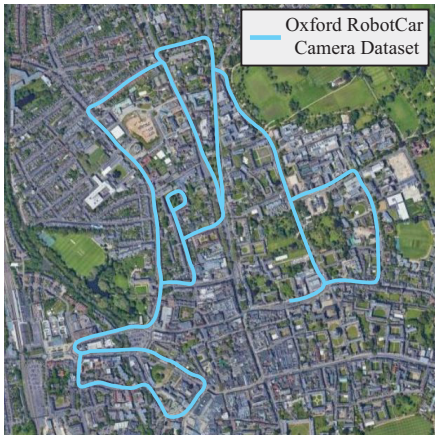

Figure 2. Oxford RobotCar map.

ments for image-based navigation and self-localization.

- The BA for image-based path planning is proposed to navigate the autonomous vehicle through the landmark identification on image-based reference query sequences.

The rest of this article is organized as follows. In Section 2, the image feature-based localization and map creation for image-based navigation are introduced. Section 3 presents the proposed BA for image-based navigation. In Section 4, the results of test cases, comparison studies, and simulations are presented, and performance characterization is provided for various maps, challenge datasets, and environments. Several important properties of the presented framework are summarized in Section 5.

2. IMAGE-BASED NAVIGATION

Image-based navigation is presented to describe our proposed nature inspired path planning algorithm. Image-based navigation is a navigation system that uses visual features and mapping to satisfy the requirements for image-based navigation and self-localization. It focuses on using reference images as a set of landmarks within a video sequence[37]. The publicly available dataset is utilized in this paper for evaluation. Figure 2 shows an aerial view of the outdoor dataset and outlines the route taken by the Oxford RobotCar from the acquired camera datasets, which will be adopted for our simulation studies.

Two sub-tasks handle compact map construction and accurate self-localization in this research:

(a) Finding of landmarks in a sub-sequence of the Oxford RobotCar run captured (see Figure 3)[37], and
(b) Matching to a short reference query sequence.

Precalculated distances are used based on features extracted from a VGG-16 + NetVLAD + whitening network. An Off-the-shelf Pitts30k model is utilized, which is freely available on the NetVLAD[29]. This solution is implemented using the NetVLAD TensorFlow. The modified implementation of their model produces the following output figures:

- Accuracy vs. distance plot of the final matching.
- Selected landmarks.
- Scatter plot of the original reference and query sequences shown in Figure 4.
- Topology of the reference sequence used for finding landmarks with network flow.

2.1. Image feature-based localization

Recently, there have been advances in camera technologies that allow them to be used in robotic applications as navigational sensors[30]. Cameras are used to provide the basic sensor input for localization, using robot pose

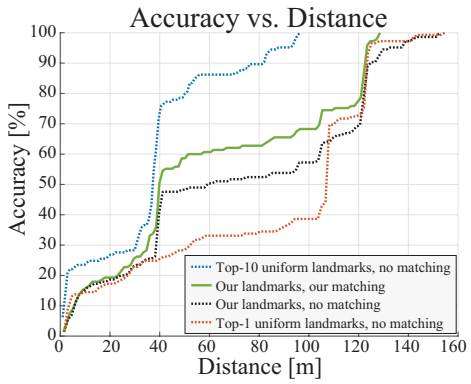

Figure 3. Accuracy vs. distance plot of the final selected landmark matching.

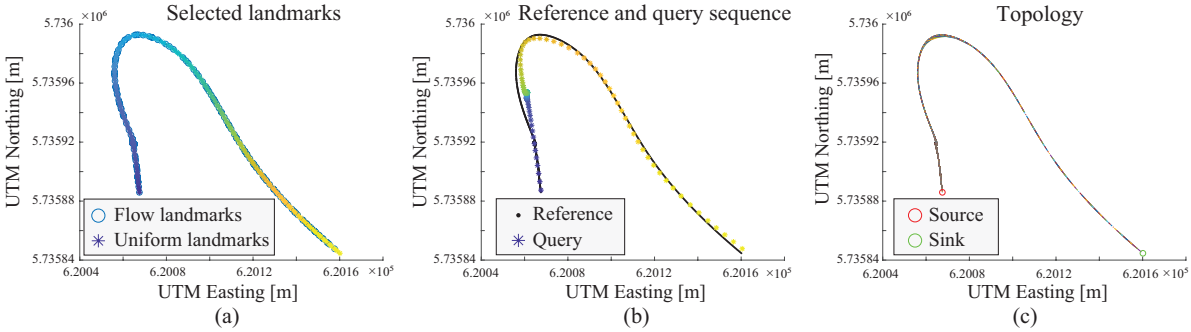

Figure 4. The scatter plot of the original reference, query sequences, and topology of reference sequence used for finding landmarks with network flow. UTM is Universal Transverse Mercator. (a) Selected landmarks. (b) Reference query. (c) Reference topology.

in each world environment[37]. Cameras deliver high quality image streams that enable feature identification, which may be employed to determine the location of the robot within the defined world environments. Images, for example, can still contain embedded information in the form of features such as points, lines, conics, spheres, and angles, which might be valuable for image-based navigation.

When images are coupled with the CNNs and appropriate filters are used to extract these features, localization can be possible. Image-based localization is a large topic that includes two main types of "worlds": those environments that are known a *priori* and others that are unknown. The method using a known environment consists of online and real-time mapping, and the latter is commonly known as SLAM[34]. SLAM methods can incorporate geometric, learned, topographical, and marker identification techniques[30]. The method considered in this study involves using a *known* environment of learned features, which will be integrated with the BA.

The method is a form of learned mapping that examine features that have been identified using two types of sources[30]. The first source is based on the VGG16 CNN coupled with off-the-shelf NetVLAD weights[29], while the second source is simply the fully connected output from the last VGG16 FC3 layer[37]. These features serve as the basis for lookup queries against the landmarks that have been established by a reference dataset of known location and geometry in this paper.

The geometric distances in the test data are calculated algorithmically and defined in the paper. The results of the query are then used to establish anchors and sensitivity values, which are used to identify the best waypoints between the identified landmarks. This network of points is best represented by a graph, representing the features where the Network Flow algorithm is then used to find the path with the lowest cost from source to

destination.

The graph is represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where utilizing cost values c_{ij} and capacity values ω_{ij} to represent the edge $e_{ij} \in \mathcal{E}$ between images $\mathcal{I}_i$ and $\mathcal{I}_j$.

$$\omega_{ij} = \eta_{\mathcal{X}} \mathbf{d}\left(\mathcal{X}_i, \mathcal{X}_j\right) \text{ and } c_{ij} = \eta_{\mathcal{F}} / \mathbf{d}\left(\mathcal{F}_i, \mathcal{F}_j\right) \tag{1}$$

where $\eta_{\mathcal{X}}$ and $\eta_{\mathcal{F}}$ are weights of geometric and visual measures, respectively. Thus, $\mathbf{d}(\mathcal{X}_i, \mathcal{X}_j)$ and $\mathbf{d}(\mathcal{F}_i, \mathcal{F}_j)$ denote the geometric and visual distances between images $\mathcal{I}_i$ and $\mathcal{I}_j$, respectively.

To generate visually comparable matches, the visual distance between landmarks $\mathcal{V}' = \{v'_i\}_{i=1}^n$ and a sequence of images $\mathcal{I}_\rho$ is utilized to calculate the flow cost rate at corresponding locations $\{x_i\}_{i=1}^n$. The flows from the source to landmarks and from query pictures to the target incur no additional cost. In addition, we offer a robust loss for feature matching with associated cost values c_{ij} and capacity values ω_{ij} defined as:

$$c_{ij} = \mathcal{L}(\boldsymbol{d}(\mathcal{F}_i, \mathcal{F}_j)), \forall e_{ij} \in \mathcal{E}_\pi; c_{ij} = 0, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_\pi \tag{2}$$

where $\mathcal{L}(.)$ is the Huber loss function. Images are matched to (at most) a single landmark, so the maximum absolute flow at each query image is limited to a value of one, thus creating the capacity constraint:

$$\omega_{ij} = q, \forall e_{ij} \in \mathcal{E}_{v'}; \omega_{ij} = 1, \forall e_{ij} \in \mathcal{E} \setminus \mathcal{E}_{v'} \tag{3}$$

with $\mathcal{E}_{v'}$ and $\mathcal{E}_\pi$ representing the directed edges spanning the partition of a bipartite graph between vertices of landmarks and image sequences, respectively. Given a radius of navigation r, and source and target vertices $\{s, t\}$, the flows $\{y_{ij}\}$ are solved for by application of Second Order Cone Programming (SOCP) to the resulting convex optimization problem:

$$\begin{aligned} min_{y_{ij}} \sum_{e_{ij} \in \mathcal{E}} c_{ij} y_{ij}, \\ 0 \leq y_{ij} \leq \omega_{ij}, \quad \forall_{e_{ij}} \in \mathcal{E}, \\ y_s = q, y_i = 0, y_T = -q, \forall v_i \in \mathcal{V} \setminus (s \cup t), \\ \| \sum_{v_i \in \mathcal{V}'} x_i y_{i(l+1)} - \sum_{v_i \in \mathcal{V}'} x_i y_{il} \| \leq r, \forall \pi_l, \pi_{l+1} \in \Pi \end{aligned} \tag{4}$$

The solution of Equation (4) is used as the location, x_l, of the query image $\mathcal{I}_l$ indicated by the vertex $\pi_l \in \Pi$.

2.2. Map creation from neural network features

The output from the NN is a feature map matrix of nodes, edges, and geographic distances between objects that correspond to actual physical locations from a small reference query (see Figure 5a). Therefore, in our developed model, a graph is created to represent the map of free space that is to be explored/traversed by our bats (Figure 6). All nodes in this graph correspond to geographic coordinates of features extracted from along the global path taken by the Oxford RobotCar (Figure 5b). These coordinates are also provided as a matrix of Northing/Easting values used in most GPS navigation systems. Section 3 provides further details on the use of the Bat Algorithm for image-based navigation in this research.

3. BAT ALGORITHM FOR IMAGE-BASED NAVIGATION

In this section, the BA for image-based path planning is proposed in consideration of the image maps obtained in Section 2.

3.1. Bat algorithm

The BA is a fresh technology inspired by the social behaviors of bats and their use of echolocation for distance sensing. It pertains to the swarm intelligence family of optimization algorithms. The BA is based on the premise that certain echolocation qualities are idealized, as specified in the following specific rules.

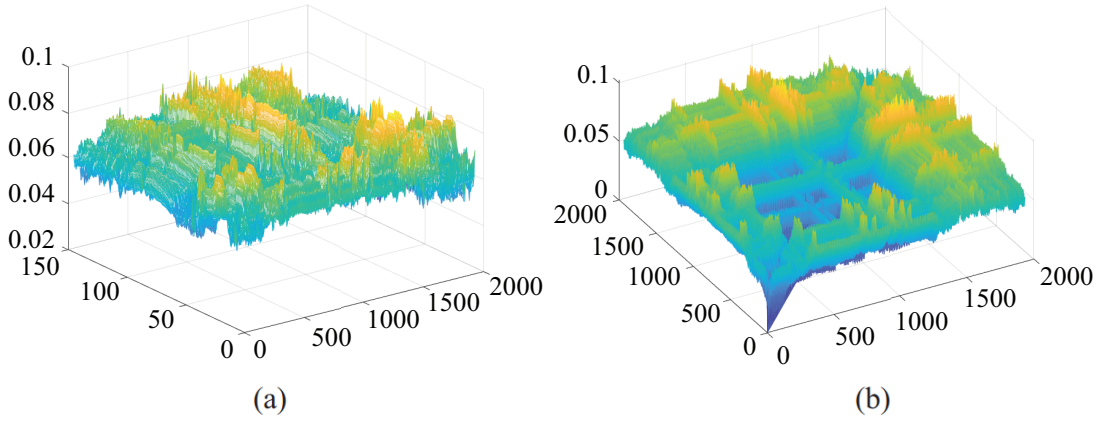

Figure 5. Neural network image feature space. (a) Matched reference sequence feature space. (b) Features space extracted from the RobotCar dataset.

- Bats apply echolocation to sense distance and always have knowledge of their surrounding environment.
- Bats fly randomly with a velocity v_i and a fixed frequency f_{min} at a position x_i, varying the wavelength λ, and loudness A_0 as they hunt for prey. They are able to instantly sense the wavelength of their emitted pulses and adjust the rate of the pulse emission $r \in [0, 1]$ based on the proximity of their target.
- The loudness of the emission varies from a minimum constant (positive) A_{min} to a large A_0.

In the BA, each bat is defined by its position x_i^t, velocity v_i^t, frequency f_i, loudness A_i^t, and the emission pulse rate r_i^t in a d-dimensional search space. The new solutions x_i^t and velocities v_i^t at time step t are given by

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min})\beta, \\ \boldsymbol{v}_i^t &= \boldsymbol{v}_i^{t-1} + (\boldsymbol{x}_i^t - \boldsymbol{x}_*) f_i, \\ \boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} + \boldsymbol{v}_i^t, \end{aligned} \tag{5}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. $\boldsymbol{x}_*$ is the current global optimized solution from among all n bats. In general, the values that are typically assigned to the frequency range of the BA fall between 0 and 100, such that $f_{min} = 0$ and $f_{max} = 100$, and initially, each bat in the population is given a frequency that is drawn uniformly from $[f_{min}, f_{max}]$.

During the search portion of the algorithm, the current best solution is used to calculate a new solution for each bat using a random walk.

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \epsilon A^t, \tag{6}$$

where $\epsilon \in [-1, 1]$ is a random number scaling factor and $A^t = \langle A_i^t \rangle$ is the average loudness of all bats at some time step t. Velocities and positions are updated in a similar manner to the standard updates in a PSO, where f_i essentially controls the pace and range of movement of bats.

Indeed, the BA can be considered a balanced combination of standard PSO and an intensive local search controlled by loudness and pulse rate. The loudness and rate of the pulse are updated as follows:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t, \\ r_i^{t+1} &= r_i^0 [1 - e^{(-\gamma t)}], \end{aligned} \tag{7}$$

where α and γ are constants set to $\alpha = \gamma = 0.9$ for the purposes of this paper.

3.2. Image-based bat algorithm path planning

Now that a primer to the BA and the computer-vision-based navigational topics have been presented, the integration method with the BA used to realize the combined algorithm is discussed in this section.

It is desirable to build an implementation that can integrate the essential features of the straightforward BA with the image-based navigation optimization technique. Instead of using external optimization libraries, such as the MOS optimization library, for selecting landmarks using visual features, our version of the BA is enhanced to include Dijkstra's Algorithm (DA) with the shortest path finding. The DA is incorporated into the BA to provide solutions for the routes created by individual bats.

Algorithm 1: Bat Algorithm for Image-based Navigation

Initialize the generation counter $t = 1$**;**
Randomly initialize population of N_P **bats and each bat** P **corresponding to a potential solution to the given problem;**
Initialize loudness A_i**, pulse frequency** Q_i**, pulse rate** r_i**, and initial velocities** $v_i (i = 1, 2, ..., N_P)$**;**
while $t < MaxGeneration$ **do**
  Generate new solutions by adjusting frequency, and updating velocities and locations/solutions per Equation (5);
  if $rand > r_i$ **then**
    Select best solution;
    Generate local solution from best;
  end
  Generate a new solution by flying randomly;
  if $rand < A_i$ *and* $f(x_i) < f(x_*)$ **then**
    Accept the new solutions;
    Increase r_i and reduce A_i;
  end
  Rank bats & find current best $\boldsymbol{x}_*$;
  $t = t + 1$;
end
Post-processing results and visualization

In algorithm 1, the BA is shown with two principal procedures defined. The first procedure is used to initialize all of the algorithm parameters required during the iterations of the computational steps, and the second procedure describes the actual computational flow. In the compute solution procedure, random adjustments to frequencies and velocities generate and select new solutions from the individual bats at any given time step. The best solution is carried forward to subsequent time steps until either the maximum iterations are reached or the optimal solution is found. The Equation (5)-Equation (7) are used for each of the operations referred to by the psuedocode presented below. Figure 6 illustrates the normalized feature coordinates in the Northing/Easting plane as nodes, and the length of the interconnecting line segments correspond to the distances between each feature illustrated in Figure 6. Thus, finding the shortest path through this network will also be the shortest path through the real geometry that is represented by the elements of the graph in this paper. By using both the graph and the coordinate matrices in this paper, a reference topological map and its coordinates can be created and then provided as arguments to the BA.

3.3. Algorithm complexity

The proposed image-based navigation involves the following steps:

(1) Capturing images of the environment;

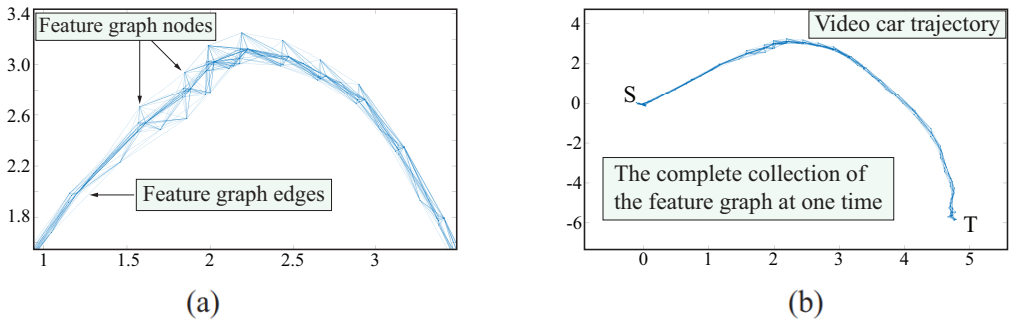

Figure 6. Feature graph nodes (spatial coordinates) and edges (geographic distance between features). (a) Closeup of feature graph. (b) Relative feature graph.

(2) Processing and extracting visual features from the images;
(3) Identifying the location of the vehicle using the extracted features;
(4) Generating a path for the vehicle based on the location information.

The time complexity of image processing and feature extraction depends on the complexity of the algorithms used, the size and resolution of the images, and the number of features extracted. For example, CNNs used for feature extraction can have a time complexity of $O(n^2)$, where n is the number of pixels in the image. However, various optimization techniques, such as down sampling and parallel processing, can be used to reduce the computational complexity.

The time complexity of BA used for image-based navigation in this paper depends on the complexity of the BA and the size of the environment. The worst-case time complexity of the BA is $O(\mathcal{N} * \mathcal{I} * \mathcal{C})$, where $\mathcal{N}$ is the number of bats, $\mathcal{I}$ is the maximum number of iterations, and $\mathcal{C}$ is the time complexity of the objective function evaluation. However, in practice, the number of iterations required is much smaller, making the algorithm computationally efficient. The time complexity of the BA is dependent on the size and complexity of the input image datasets (landmarks) and the convergence rate of the algorithm. Besides, the BA is known for its efficiency and effectiveness in solving complex optimization problems.

The time complexity of the proposed image-based navigation is dependent on the properties of the environment being explored. The BA, as an efficient algorithm and optimization technique, is used to reduce the computational complexity and improve the performance of image-based navigation.

4. SIMULATION AND COMPARISON STUDIES

In this section, simulation and comparison studies are undertaken to demonstrate the effectiveness, feasibility, and robustness of our proposed BA path planning method. In the first experiment, simulations are conducted by utilizing different maps, and the results are compared with other state-of-the-art path planning algorithms. In the second experiment, the proposed BA method is applied to the map of real-world scenarios with random simulated image-based landmarks. In the third experiment, we apply this algorithm to image-based navigation in real environments (The image-based datasets are taken by the Oxford RobotCar).

4.1. Simulation and comparison studies in various environments

To evaluate our proposed BA model, we first use it within a grid map and compare it with the PSO algorithm, Fuzzy NN (FNN), and Hybrid PSO and FNN Algorithm (HPFA) from [38] in Scenario 1. Two grid-based maps are selected in Figure 7 and Figure 8. The size of the map is 30×30. The population size of BA is set to 50. The parameter settings and experimental results of other algorithms are in [38]. The comparative results of a variety of path planning methods are illustrated in Figure 7, which demonstrates that all four algorithms

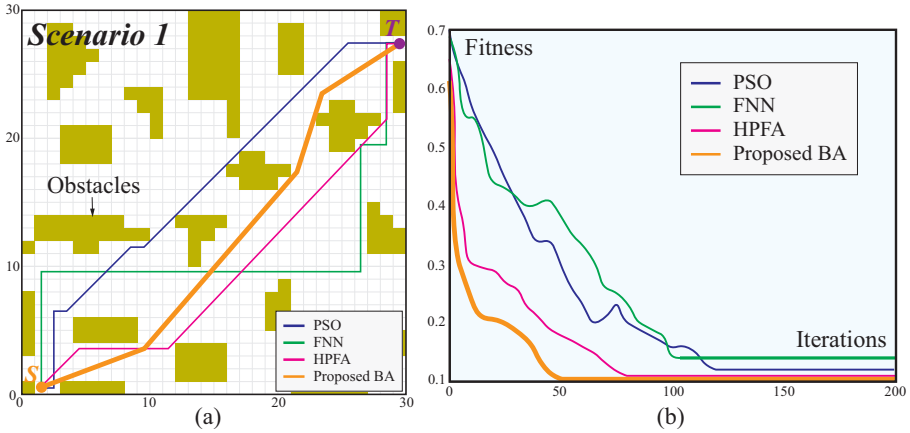

Figure 7. Path planning results in Scenario 1 from [38]. (a) Generated trajectories of PSO, FNN, HPFA, and proposed BA. (b) Convergence of PSO, FNN, HPFA, and proposed BA.

Table 1. Comparison of maximum path length, minimum path length, and average processing time of PSO, FNN, HPFA, and proposed BA

Algorithm	Max length	Min length	Average time (ms)
PSO	85.71	46.97	1639
FNN	102.89	48.28	1921
HPFA	74.76	40.76	*1426*
Proposed BA	**42.56**	**33.62**	**1597**

can effectively plan the way from the start to the target points in the grid map environment. However, the trajectory planned by the proposed BA is much shorter than the paths generated by the other three techniques. Table 1 summarizes the qualitative comparison between the features of our algorithm and selected algorithms, including maximum path length, minimum path length, and average processing time.

The simulation results in terms of the convergence of various methods are shown in Figure 7(b). The fitness function converges to a pretty high value after around 100 iterations of the PSO algorithm, as shown by the findings, which indicate that the method first converges more quickly and subsequently slows down significantly. The FNN method is distinct in that its convergence speed increases, and it converges after around 120 iterations; nevertheless, the fitness function converges to a higher value than the PSO algorithm. The HPFA method has a relatively better convergence result, while its convergence speed is slow. After around 50 iterations, the proposed BA method converges to the lowest fitness function among the four methods, and thus the convergence curve turbulence is minor.

In order to further verify the optimization and path planning capabilities of the proposed BA, we conducted another grid-based environment comparison study with Ant Colony Optimization (ACO), Genetic Algorithm (GA), A* Algorithm Optimization (AAO), Kth Shortest Path Algorithm (KSPA), and HPFA in Scenario 2. The simulation results provided in Figure 8(a) reveal that the path length of the proposed BA algorithm is substantially smoother and shorter than the other five methods. Figure 8(b) summarizes the length findings of the trajectories. When compared to previous grid map path planning algorithms, our developed BA approach has superior convergence and optimal solution. This might highlight the viability of the proposed BA in path planning.

To validate the adaptability and efficiency of our algorithm in various environments, one map with resolution 1500×1500 from Massachusetts Roads Dataset [56] is then selected for simulation and comparative studies. The initial and multiple target positions of the autonomous vehicle are randomly set. The initial position is

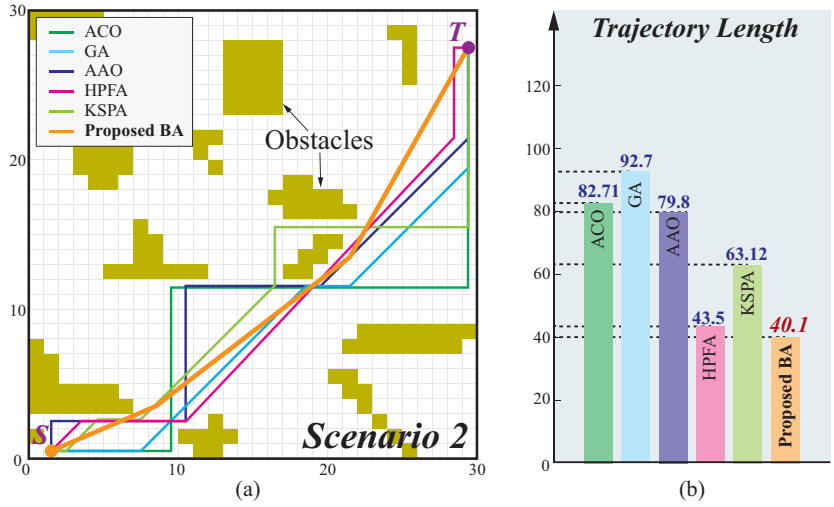

Figure 8. Path planning results in Scenario 2 from [38]. (a) Generated trajectories of ACO, GA, AAO, HPFA, KSPA, and proposed BA. (b) Trajectories length of ACO, GA, AAO, HPFA, KSPA, and proposed BA.

(1108, 1433), target positions are $T_1 = (189, 1062)$, $T_2 = (381, 163)$, $T_3 = (1431, 640)$, and $T_4 = (728, 611)$, as shown in Figure 9. Two cutting-edge path planning algorithms, Probabilistic Roadmap (PRM) and Rapidly Exploring Random Tree* (RRT*), are utilized for comparative studies. PRM is a well-known and effective path planning method based on sampling. This approach can seek a solution utilizing a limited number of random sampling points and takes minimal processing time. It generates random sampling points in the free space of a given workspace to create a route network graph. Then, use Dijkstra to search the created route network graph for suitable routes. The number of PRM sampling points in this study is 1500. RRT* is also a sampling-based path planning algorithm similar to PRM. From the initial location, it randomly generates the sampling point to the spanning tree in the workspace and links it to the nearest obstacle-free point on the path tree to the sampling point. In all situations, the maximum RRT* iteration time is set to 500000, and the maximum connection distance is set to 4. We repeatedly execute 40 times in the selected scenario. Table 2 provides a qualitative assessment of the path length, smoothness rate, and execution time of the algorithm in comparison to other algorithms. The path smoothness rate ψ is based on the sum of the angles differences between adjacent path segments calculated by Equation (8).

$$\psi = \sum_{i=2}^{n-1} \text{abs}\left(\theta_{i+1} - \theta_i\right) \tag{8}$$

$$\theta_{i+1} = \text{atan}\left[\left(y_{i+1} - y_i\right) / \left(x_{i+1} - x_i\right)\right] \tag{9}$$

$$\theta_i = \text{atan}\left[\left(y_i - y_{i-1}\right) / \left(x_i - x_{i-1}\right)\right] \tag{10}$$

The results indicate that path planning methods, such as PRM and RRT*, are insufficient when the sample points are inadequate or the distribution of the map is inappropriate. Although the RRT* method has a lower execution time, the proposed BA approach has a significant advantage in terms of path length and smoothness.

4.2. Image-based navigation with simulated datasets

Image-based navigation utilizes reference pictures as a collection of landmarks in a video sequence to meet image-based navigation and self-localization navigation techniques. However, due to insufficient datasets, the

Table 2. Comparison of path length, path smoothness rate, algorithm execution time, and success rate of PRM, RRT*, and the proposed Bat Algorithm. These values are reported as the result of 40 executions

$S \Longrightarrow T$	Model name	Minimum path length	Average path length	Smoothness rate (rad)	Average execution time (sec)	Success rate (%)
$S \Longrightarrow T_1$	PRM	1251.3	1690.5	1.0099	598.07	67.5
	RRT*	1384.8	144.88	0.6701	**30.84**	95
	BA	**1073.5**	**1246.2**	**0.2133**	329.65	**100**
$T_1 \Longrightarrow T_2$	PRM	1287.4	1403.8	1.1263	482.10	77.5
	RRT*	1505.3	1589.0	0.9239	**18.63**	**100**
	BA	**993.44**	**1079.4**	**0.1357**	134.72	**100**
$T_2 \Longrightarrow T_3$	PRM	1507.9	1540.1	0.7502	500.02	72.5
	RRT*	1402.0	1448.8	0.6137	**27.13**	92.5
	BA	**1252.4**	**1246.2**	**0.2133**	329.65	**100**
$T_3 \Longrightarrow T_4$	PRM	1292.3	1318.6	0.2328	494.10	62.5
	RRT*	825.94	884.39	0.5306	**35.66**	95
	BA	**735.48**	**801.27**	**0.2133**	329.65	**100**

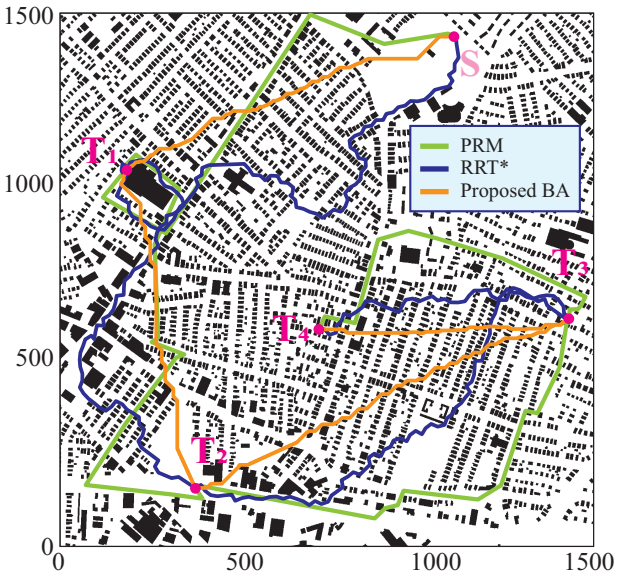

Figure 9. Illustration of the real-world map from Massachusetts Roads Dataset, composed of 1500×1500 nodes. The pink circle marks are starting and target points. The green line, blue line, and orange line denote the trajectory of PRM, RRT*, and our proposed BA, respectively.

navigation experiment is conducted by utilizing simulated images as landmarks in this paper. It should be noted that since the reference image or the captured video is dependent extremely on the road segment passed by the previous video car, its generated landmarks are limited and random in the environment. The image-based landmarks we simulated have the same limitations, being only within a random portion of the map [Figure 10]. Therefore, to evaluate our proposed BA model, the image-based landmarks that are simulated have the same limitations, being only within a random portion of the map (Figure 10). Two algorithms, Dijkstra's algorithm and Slime Mound Algorithm (SMA)[57], are selected for comparison. The test workspace is a real-world map, as New York City map from the benchmark[58]. 100 landmarks and 150 landmark data are simulated in Figure 10(a) and Figure 10(b), respectively. Among them, in Figure 10(a) and Figure 10(b), the green and yellow lines indicate the feasibility of connecting between landmarks (spatial coordinates) in the topological map, which are geographic distances between features. Purple, pink, and orange lines in Figure 10(a) and Figure 10(b) represent the final trajectories of Dijkstra's algorithm, SMA, and the proposed BA, respectively. Table 3 outlines the length of the trajectory planned by the tested algorithms. The results demonstrate that in the complex connection topological map, our proposed BA algorithm can obtain shorter paths than the Dijkstra's algorithm and SMA.

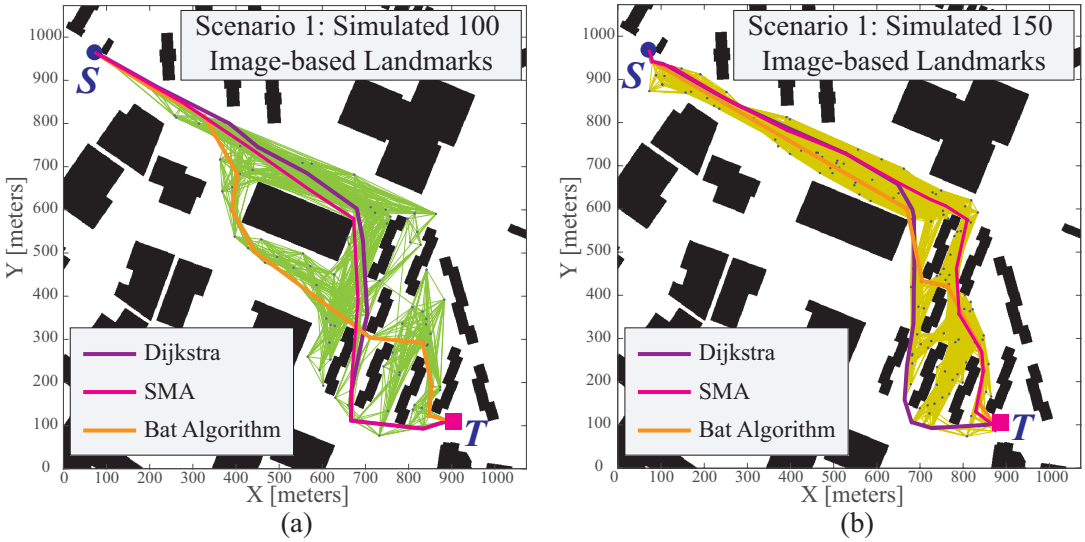

Figure 10. Comparison of the trajectory by the proposed Bat Algorithm image-based navigation with basic Dijkstra's algorithm. The image-based landmark datasets are randomly generated in the New York benchmark map [58].

Table 3. Comparison of the Dijkstra's algorithm and SMA with proposed BA in simulated image-based landmarks datasets

Scenario	Simulated image-based landmarks	Algorithm	Path length
Figure 10(a)	100	Dijkstra	775.1
		SMA	768.7
		Proposed BA	**710.9**
Figure 10(b)	150	Dijkstra	771.2
		SMA	743.9
		Proposed BA	**700.2**

4.3. Image-based navigation with Oxford RobotCar Datasets

The simulation and comparison studies in this section aim to validate the proposed image-based BA navigation in real-world datasets. The image-based datasets are taken by the Oxford RobotCar. The proposed BA path planner initially obtains the solid green trajectory with the real world maps, as illustrated in Figure 11. While the updated dataset finds that the road segment is interrupted by an obstacle (as a blue polygon-shaped obstacle in Figure 11), BA can replan the trajectory in light of the existing dataset. The dashed purple and pink trajectories reveal the proposed image-based BA navigation with the replanned procedure. The detail of the proposed image-based BA navigation is accomplished by modifying and building the BA algorithm with an integrated Dijkstra's algorithm. The input to the BA algorithm is the graph representation of the reference query sequence shown in Figure 6 and a matrix of indices that corresponds to the feature coordinates.

Once the BA algorithm is functional, it is converted and integrated at the correct location to process the image-feature graph. Output compares with the Thoma's output [37]. The BA generated a series of graphs that are superimposed on the images presented in Section 3. Figure 12 also displays the best-fit values of each BA iteration to illustrate how swiftly the BA converges to its optimal solution. Several simulations were carried out utilizing our integrated technique, and the findings shown here are illustrative of typical worst-case convergence times. Each simulation result is accompanied by a brief caption and a zoomed-in view, allowing the reader to gain a clearer sense of the volume and scale of each of the features being processed in the graph. The intention is to demonstrate how the BA output compares to the Thoma's output [37].

As can be seen in Figure 12 above, the BA was able to converge upon an optimal solution in 33 iterations. This represents the worst-case results observed during the experimentation phase of this project. On average, convergence occurs in approximately 5-10 iterations in our studies. In the results shown in Figure 13 and Figure 14, the BA output is overlayed against the original simulation outputs from the [37]. It is clear from these

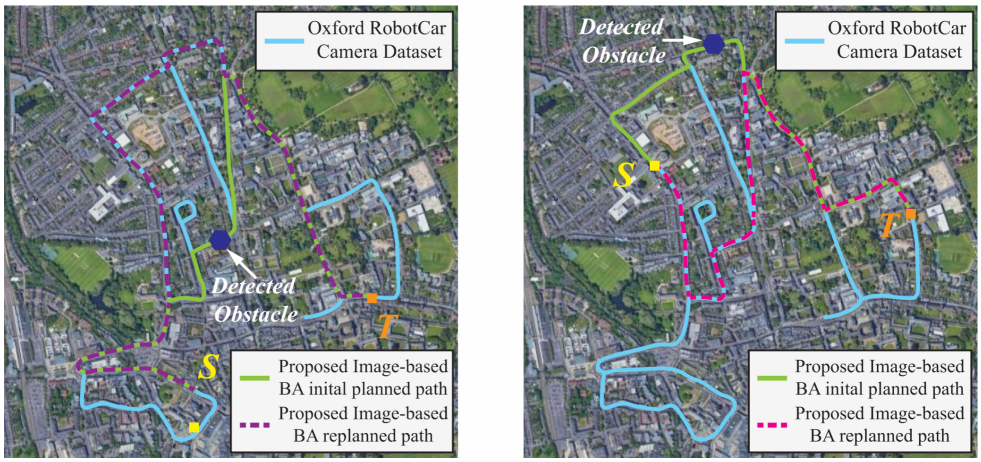

Figure 11. Illustration of the overall trajectory by the proposed image-based Bat Algorithm navigation. The image-based datasets are taken by the Oxford RobotCar.

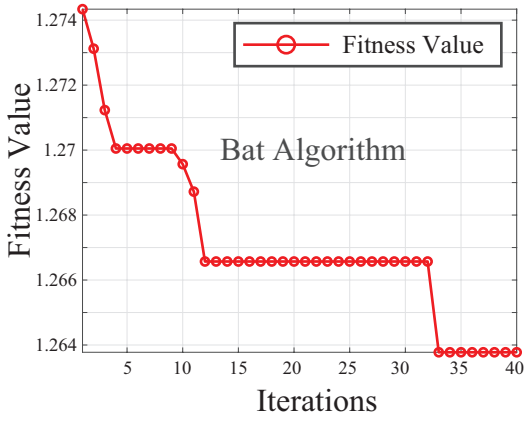

Figure 12. Number of iterations until Bat Algorithm convergence.

plots that the BA finds the optimal path through the given topography and does not hinder or otherwise alter the effectiveness of the global navigation strategy.

4.4. Analysis and discussions

Several simulations using our developed hybrid model were performed during the course of testing this implementation, with the results of the BA proving to be adept at traversal of the known topography produced by the image feature-based recognition. This is an encouraging result since new feature identification strategies are the topic of much research today and are likely to achieve even better results over the next few years. The true test of whether or not the algorithm can be used in real world environments will come once a dynamic obstacle avoidance method can be added to the algorithm.

We made many attempts to integrate the BA algorithm at various locations in the processing pipeline, but it became clear that the geographic feature distances were not *calculated in* the [37] algorithm but were instead *processed by* the algorithm. This makes the identification of the best place to perform the task of integration very difficult and may point to the need to have optimization algorithms performed up front in the process, potentially eliminating the need for other proposed methods entirely (future work). Nevertheless, the best implementation of BA would take place where the path-optimizer and the graph-traversal libraries can be switched out with their equivalent optimization biologically inspired implementation.

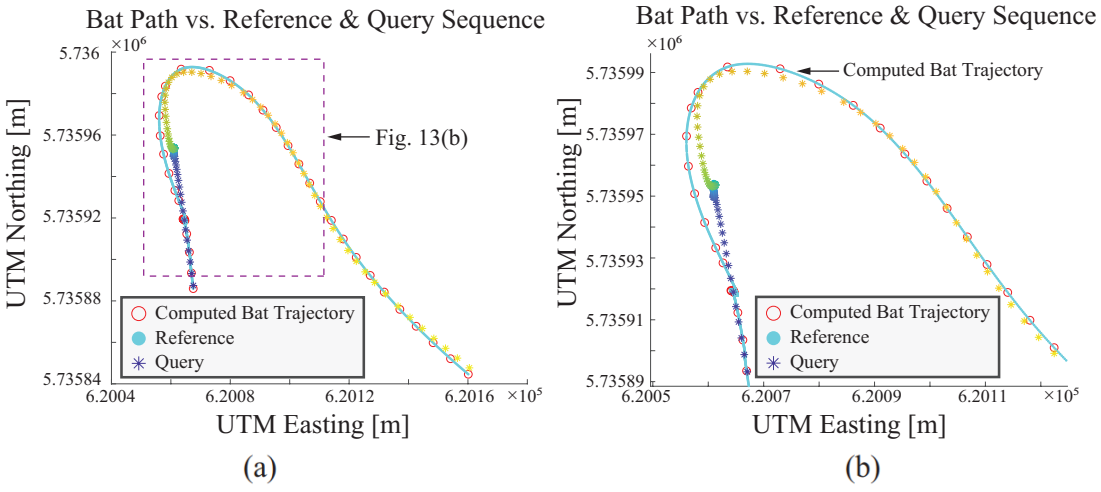

Figure 13. Reference query with Bat Algorithm solution superimposed. UTM is Universal Transverse Mercator. (a) Bat Algorithm solution vs. reference query. (b) Zoomed solution vs. reference query.

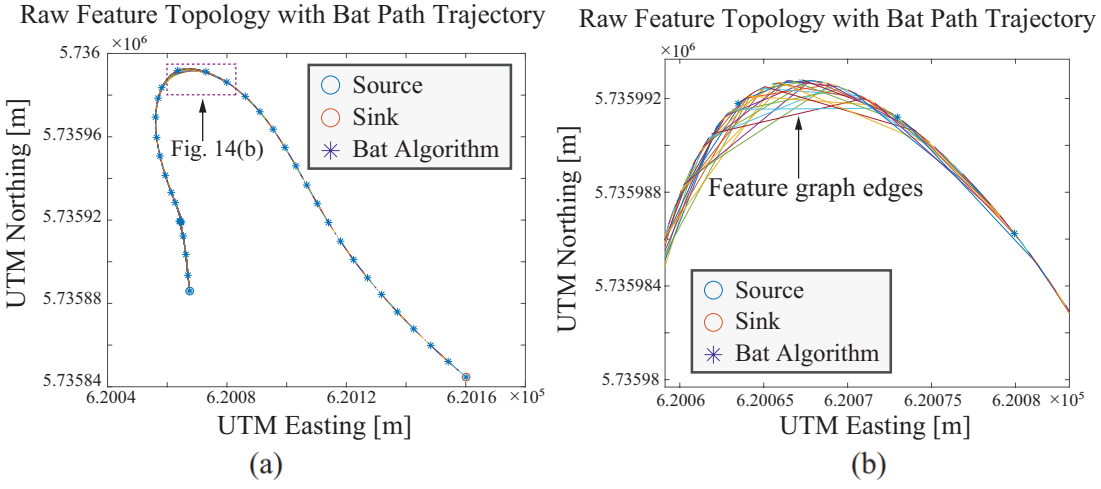

Figure 14. Raw feature topology with Bat Algorithm solution superimposed. (a) Bat Algorithm solution vs. raw feature topology. (b) Zoomed Bat Algorithm solution vs. raw feature topology.

The full image feature extraction process was explored here to process the raw image features into the format that is best suited to a dynamic real-time path planner with inherent obstacle avoidance such as the D*-Lite algorithm [59]. Thoma *et al.* [37] integrated a localization method against a reference map, but to be fully integrated with a path planning algorithm, it needs to be more functional in a real-world sense. Other researchers have focused on competing technologies, such as using PSO methodology to train FNNs [38] focusing on the optimization of parameters during the front-end NN feature extraction. Yet, others have used reinforcement learning techniques to select an optimal path in ultra-dynamic emergency environments through hybrid NNs and A* algorithms [54]. Their reasoning is to focus entirely on the back-end processing to help intelligent vehicles plan dynamic routes around traffic emergency conditions, including limitations on vehicle height, width, weight restrictions, or accidents and traffic jams.

These approaches differ from the current research in that they focus on optimizing both before and long after map building occurs. The need for a simultaneous SLAM-like solution remains. When compared with other biologically inspired algorithms for path planning in image-based systems, the BA has shown that it is capable

of the challenge, but it will also need to prove that it can be deployed in a mature and accurate localizing methodology (which has been demonstrated on a known dataset, but not on a live video camera feed). For these reasons, it would make the most sense to integrate a parallel architecture with the path planner that has access to those same raw features as the SLAM method given in the article. The local map that is built by the raw camera feed is used to generate a current coordinate location for the autonomous vehicle body within the region of the visible and identified map. The path planning algorithm, however, would be capable of resolving a path to the goal location from the current localized coordinates up to the extreme edge of the map, which could presumably be incomplete or inaccurate due to the sensor errors. A heuristic distance to the end goal coordinate could be used as a guideline to navigate in the local region, avoiding and replanning based on dynamic obstacle avoidance, and as new information becomes available in the dataset, continue to work toward that goal location.

In future research, an experiment and design to implement the BA into a Field Programmable Gate Array (FPGA) would also help to keep the burden of constant re-calculation down and help to distribute the task of feature extraction to a specific piece of parallel modular hardware. An interface to the input camera feed can be sent to the SLAM module and, simultaneously, to the embedded hardware housing the FPGA.

5. CONCLUSION

Despite the potentiality of image-based navigation for the usage of an autonomous vehicle in an unknown environment, current studies on system development focused largely on the algorithms for mapping, feature retrievals, and localization. However, these studies in this paper concluded that selecting a robust, reliable, and efficient path planning algorithm was at least as important to the success of an image-based navigation system.

The goal of the presented work was to develop an original Bat path planning algorithm. By introducing the BA at a point where the graph of features is calculated, we were able to show how a biologically inspired algorithm can be used to enhance the SLAM performance with a local path planner. The results presented here show that the BA can produce path planning results that are not only optimized but also maintain the integrity of the image-based feature recognition and self-localization upon which it has been added.

It was our finding that during the overall integration of the BA into the image-based system framework, it became apparent that there is an increased need to gain direct access to the raw image features as they are extracted from the NNs themselves. Although it was the original intent of this paper to work within the given framework, it became clear that the computational expense of this methodology is high. Hence, for our future exploration, it should be prudent to modify the research parameters to implement a more parallel approach to the BA that aided in the path planning by simultaneously sharing information between the two systems. This type of solution would also benefit from having FPGA hardware integrated that could perform the tasks of DNNs in real time and assist in distributing those features to the BA and the SLAM algorithm in parallel.

DECLARATIONS

Acknowledgments

The authors would like to thank the editor-in-chief, the associate editor, and the anonymous reviewers for their valuable comments.

Authors' contributions

Made substantial contributions to the research, idea generation, algorithm design, and simulation and wrote and edited the original draft: Short D, Lei T, Carruth D, and Luo C

Performed critical review, commentary, and revision and provided administrative, technical, and material support: Carruth D, Bi Z

Financial support and sponsorship
This research was supported by the Mississippi Space Grant Consortium under the NASA EPSCoR RID grant.

Availability of data and materials
Not applicable.

Conflicts of interest
All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

REFERENCES
1. Wang L, Luo C, Li M, Cai J. Trajectory planning of an autonomous mobile robot by evolving ant colony system. *Int J Robot Autom* 2017;32:406–13. DOI
2. Lei T, Chintam P, Carruth DW, Jan GE, Luo C. Human-autonomy teaming-based robot informative path planning and mapping algorithms with tree search mechanism. In: 2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS). IEEE; 2022. pp. 1–6. DOI
3. Zhao W, Lun R, Gordon C, et al. A privacy-aware Kinect-based system for healthcare professionals. In: IEEE International Conference on Electro Information Technology (EIT); 2016. pp. 0205–10. DOI
4. Jayaraman E, Lei T, Rahimi S, Cheng S, Luo C. Immune system algorithms to environmental exploration of robot navigation and mapping. In: Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12. Springer; 2021. pp. 73–84. DOI
5. Lei T, Chintam P, Luo C, Rahimi S. Multi-robot directed coverage path planning in row-based environments. In: 2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). IEEE; 2022. pp. 114–21. DOI
6. Zhu D, Yan T, Yang SX. Motion planning and tracking control of unmanned underwater vehicles: technologies, challenges and prospects. *IR* 2022;2:200–22. DOI
7. Li J, Xu Z, Zhu D, et al. Bio-inspired intelligence with applications to robotics: a survey. *IR* 2021;1:58–83. DOI
8. Lei T, Luo C, Ball JE, Rahimi S. A graph-based ant-like approach to optimal path planning. In: IEEE Congress on Evolutionary Computation (CEC); 2020. pp. 1–6. DOI
9. Chen J, Luo C, Krishnan M, Paulik M, Tang Y. An enhanced dynamic Delaunay triangulation-based path planning algorithm for autonomous mobile robot navigation. In: Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques. vol. 7539. SPIE; 2010. pp. 253–64. DOI
10. Sellers T, Lei T, Luo C, Jan GE, Ma J. A node selection algorithm to graph-based multi-waypoint optimization navigation and mapping. *IR* 2022;2:333–54. DOI
11. Lei T, Sellers T, Luo C, Zhang L. A bio-inspired neural network approach to robot navigation and mapping with nature-inspired algorithms. In: International Conference on Swarm Intelligence. Springer; 2022. pp. 3–16. DOI
12. Luo C, Yang SX, Meng MQH. Neurodynamics based complete coverage navigation with real-time map building in unknown environments. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2006. pp. 4228–33. DOI
13. Sellers T, Lei T, Jan GE, Wang Y, Luo C. Multi-objective optimization robot navigation through a graph-driven PSO mechanism. In: Advances in Swarm Intelligence: 13th International Conference, ICSI 2022, Xi'an, China, July 15–19, 2022, Proceedings, Part II. Springer; 2022. pp. 66–77. DOI
14. Jan GE, Luo C, Hung LP, Shih ST. A computationally efficient complete area coverage algorithm for intelligent mobile robot navigation. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE; 2014. pp. 961–66. DOI
15. Lei T, Luo C, Ball JE, Bi Z. A hybrid fireworks algorithm to navigation and mapping. In: Handbook of Research on Fireworks Algorithms and Swarm Intelligence. IGI Global; 2020. pp. 213–32. DOI
16. Li X, Li X, Khyam MO, Luo C, Tan Y. Visual navigation method for indoor mobile robot based on extended BoW model. CAAI Transactions on Intelligence Technology 2017;2:142–47. DOI
17. Lei T, Luo C, Jan GE, Bi Z. Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm. *Front Robot AI* 2022;9. DOI

18. Wang D, Yang SX. Intelligent feature extraction, data fusion and detection of concrete bridge cracks: current development and challenges. *IR* 2022;2:391–406. DOI
19. Lei T, Li G, Luo C, et al. An informative planning-based multi-layer robot navigation system as applied in a poultry barn. *IR* 2022;2:313–32. DOI
20. Ortiz S, Yu W. Autonomous navigation in unknown environment using sliding mode SLAM and genetic algorithm. *IR* 2021;1:131–50. DOI
21. Lei T, Sellers T, Rahimi S, Cheng S, Luo C. A nature-inspired algorithm to adaptively safe navigation of a Covid-19 disinfection robot. In: International Conference on Intelligent Robotics and Applications. Springer; 2021. pp. 123–34. DOI
22. Luo C, Gao J, Murphey YL, Jan GE. A computationally efficient neural dynamics approach to trajectory planning of an intelligent vehicle. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE; 2014. pp. 934–39. DOI
23. Liu L, Luo C, Shen F. Multi-agent formation control with target tracking and navigation. In: IEEE International Conference on Information and Automation (ICIA); 2017. pp. 98–103. DOI
24. Zhao W, Lun R, Gordon C, et al. Liftingdoneright: a privacy-aware human motion tracking system for healthcare professionals. *IJHCR* 2016;7:1–15. DOI
25. Lei T, Luo C, Jan GE, Fung K. Variable speed robot navigation by an ACO approach. In: International Conference on Swarm Intelligence. Springer; 2019. pp. 232–42. DOI
26. Luo C, Yang SX, Mo H, Li X. Safety aware robot coverage motion planning with virtual-obstacle-based navigation. In: 2015 IEEE International Conference on Information and Automation. IEEE; 2015. pp. 2110–15. DOI
27. Luo C, Yang SX, Krishnan M, Paulik M. An effective vector-driven biologically-motivated neural network algorithm to real-time autonomous robot navigation. In: IEEE International Conference on Robotics and Automation (ICRA); 2014. pp. 4094–99. DOI
28. Zhu D, Tian C, Jiang X, Luo C. Multi-AUVs cooperative complete coverage path planning based on GBNN algorithm. In: 29th Chinese Control and Decision Conference (CCDC); 2017. pp. 6761–66. DOI
29. Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 5297–307. DOI
30. Maltar J, Marković I, Petrović I. Visual place recognition using directed acyclic graph association measures and mutual information-based feature selection. *Rob Auton Syst* 2020;132:103598. DOI
31. Cao F, Liu B, Park DS. Image classification based on effective extreme learning machine. *Neurocomputing* 2013;102:90–97. DOI
32. Muthusamy H, Polat K, Yaacob S. Improved emotion recognition using gaussian mixture model and extreme learning machine in speech and glottal signals. *Math Probl Eng* 2015;2015. DOI
33. Yang Y, Deng Q, Shen F, Zhao J, Luo C. A shapelet learning method for time series classification. In: IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI); 2016. pp. 423–30. DOI
34. Cieslewski T, Choudhary S, Scaramuzza D. Data-efficient decentralized visual SLAM. In: 2018 IEEE International Conference on Robotics and Automation (ICRA); 2018. pp. 2466–73. DOI
35. Oishi S, Inoue Y, Miura J, Tanaka S. SeqSLAM++: View-based robot localization and navigation. *Rob Auton Syst* 2019;112:13–21. DOI
36. Otte M. A survey of machine learning approaches to robotic path-planning. University of Colorado at Boulder; 2008. PhD Preliminary Exam.
37. Thoma J, Paudel DP, Chhatkuli A, Probst T, Gool LV. Mapping, localization and path planning for image-based navigation using visual features and map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 7383–91. DOI
38. Liu X, Zhang D, Zhang J, Zhang T, Zhu H. A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. *Cluster Comput* 2021;24:1901–15. DOI
39. Zhang X, Yang Z, Cao F, et al. Conditioning optimization of extreme learning machine by multitask beetle antennae swarm algorithm. *Memet ComputMemet Comput* 2020;12:151–64. DOI
40. Chu Z, Wang F, Lei T, Luo C. Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. *IEEE Trans Intell Veh* 2023;8:108–20. DOI
41. Dai P, Taghia J, Lam S, Katupitiya J. Integration of sliding mode based steering control and PSO based drive force control for a 4WS4WD vehicle. *Auton Robot* 2018;42:553–68. DOI
42. Teng F, Zhang H, Luo C, Shan Q. Delay tolerant containment control for second-order multi-agent systems based on communication topology design. *Neurocomputing* 2020;380:11–19. DOI
43. Chu Z, Sun B, Zhu D, Zhang M, Luo C. Motion control of unmanned underwater vehicles via deep imitation reinforcement learning algorithm. *IEEE trans Intell Transp Syst* 2020;14:764–74. DOI
44. Shishika D, Paley DA. Mosquito-inspired distributed swarming and pursuit for cooperative defense against fast intruders. *Auton Robot* 2019;43:1781–99. DOI
45. Lei T, Luo C, Sellers T, Rahimi S. A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping. *Intelligent Systems with Applications* 2021;12:200053. DOI
46. Zhao QS, Hu YL. Multidimensional scaling localisation algorithm based on bacterial colony chemotaxis optimisation. *IEEE Trans Mob Comput* 2016;11:151. DOI
47. Pham DT, Castellani M. A comparative study of the bees algorithm as a tool for function optimisation. *Cogent Eng* 2015;2:1091540. DOI
48. Li X, Xiao S, Wang C, Yi J. Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem. *Memet Comput* 2019;11:371–89. DOI
49. Tu D, Wang E, Zhang F. An intelligent wireless sensor positioning strategy based on improved bat algorithm. In: 2019 International

Conference on Intelligent Transportation, Big Data & Smart City (ICITBS); 2019. pp. 174–77. DOI
50. Wang GG, Chu HE, Mirjalili S. Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerosp Sci Technol* 2016;49:231–38. DOI
51. Lei T, Luo C, Sellers T, Wang Y, Liu L. Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots. *IEEE Trans Aerosp Electron Syst* 2022;58:5129–40. DOI
52. Chu Z, Xiang X, Zhu D, Luo C, Xie D. Adaptive trajectory tracking control for remotely operated vehicles considering thruster dynamics and saturation constraints. *ISA Trans* 2020;100:28–37. DOI
53. Zhang Q, Luo R, Zhao D, Luo C, Qian D. Model-free reinforcement learning based lateral control for lane keeping. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE; 2019. pp. 1–7. DOI
54. Liu X, Zhang D, Zhang T, et al. Novel best path selection approach based on hybrid improved A* algorithm and reinforcement learning. *Appl Intell* 2021;51:9015–29. DOI
55. Yang J, Chai T, Luo C, Yu W. Intelligent demand forecasting of smelting process using data-driven and mechanism model. *IEEE Trans Ind Electron* 2018;66:9745–55. DOI
56. Mnih V. Machine learning for aerial image labeling. University of Toronto; 2013.
57. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. *FGCS* 2020;111:300–323. DOI
58. Sturtevant NR. Benchmarks for grid-based pathfinding. *IEEE T COMP INTEL AI* 2012;4:144–48. DOI
59. Koenig S, Likhachev M. Fast replanning for navigation in unknown terrain. *IEEE Trans Robot* 2005;21:354–63. DOI