Improving Walking Path Generation Through Biped Constraint in Indoor Navigation System for Visually Impaired Individuals

Qingquan Na, Hui Zhou, Hailei Yuan, Mengfan Gui, and Hongjing Teng

***Abstract*— This paper introduces a walking path generation method specifically developed for the Smart Cane, which is a RNA (Robotic Navigation Assistance Device) aimed at enhancing indoor navigation for visually impaired individuals. The proposed approach combines the utilization of a LIPM (Linear Inverse Pendulum Model) and LFPC (Linear Foot Placement Controller) motion primitives to generate walking paths specifically designed for visually impaired individuals. The primary objective is to generate paths that conform to human motion constraints, thereby guaranteeing an efficient and natural navigation experience. Integrating autonomous navigation framework, the Smart Cane facilitates safe and effective guidance for visually impaired participants in the indoor environments. Furthermore, comparative experiments have been conducted to validate the effectiveness of the proposed method, providing evidence of its capability to generate walking paths that conform to human motion constraints. The experiment results indicate that the proposed walking path generation method is a promising solution to enhance the navigation experience of visually impaired individuals.**

***Index Terms*— Robotic navigation assistance device, visually impaired, path generation, kinodynamic planning, navigation system.**

I. Introduction

MORE than 250 million people worldwide suffer from vision impairment, ranging from moderate impairment to blindness, significantly impacting their physical health, mental well-being, and overall quality of life [1].

Vision impairment has multifaceted consequences, notably impacting the mobility of individuals to varying extents. Visual impairment often results in individuals voluntarily reducing their walking speed during travel, thereby affecting their efficiency and increasing the risk of accidental injury. Furthermore, due to mobility restrictions imposed by vision impairment and the absence of sufficient assistive tools, individuals with visual impairments are compelled to decrease their frequency of travel [1]. A survey [2] conducted in China regarding the weekly travel habits of individuals with blindness revealed that around 30% of them rarely go outdoors, while only 9% of limited vision individuals travel alone more than four times a week. To address these limitations, various RNA (Robotic Navigation Assistance Devices) [3] have been developed to assist individuals with visual impairments.

In this paper, our primary focus is on addressing the indoor navigation challenges faced by visually impaired individuals. Our objective is to investigate the potential of employing RNA to assist visually impaired individuals in navigating indoor environments. In comparison to outdoor environments, indoor environments pose more substantial challenges for navigation due to various factors:

(1) Higher obstacle density [4], [5]: Indoor spaces have a greater concentration of obstacles, including furniture and walls, which poses challenges for visually impaired individuals to navigate safely.

(2) Hanging obstacles [4], [5]: Indoor environments may have hanging objects such as light fixtures or signage, which can present additional obstacles that require spatial awareness and obstacle detection.

(3) Lack of GPS positioning [6]: Unlike outdoor environments, indoor spaces typically lack reliable GPS signals, making traditional navigation methods based on satellite signals ineffective. Thus, GPS-denied position algorithm is needed for indoor navigation.

With the advancement of autonomous navigation systems [7] and SLAM (Simultaneous Localization and Mapping) technology [8], [9] possibilities have been opened up for the development of navigation frameworks for the visually impaired population to address the aforementioned issues.

Building upon the aforementioned technological advancements, we have developed the Smart Cane, an intelligent assistive device integrated with numerous sensors and feedback mechanisms. In Figure 1, we provide the demonstration of the Smart Cane in guiding a visually impaired individual within a real-world environment. We have designed two

Manuscript received 3 August 2023; revised 2 January 2024 and 27 February 2024; accepted 6 March 2024. Date of publication 11 March 2024; date of current version 18 March 2024. This work was supported in part by Space Medical Experiment Project of China Manned Space Program under Grant HYZHXM01010; in part by the National Natural Science Foundation of China under Grant U1613228; and in part by the Open Foundation of Key Laboratory of Biorheological Science and Technology (Chongging University), Ministry of Education under Grant CQKLBST-2020-005. *(Corresponding author: Hui Zhou.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Nanjing Brain Hospital Institutional Review Board under Application No. 2023-KY116-01.

The authors are with the School of Automation, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China (e-mail: zhouhui@njust.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2024.3375944, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2024.3375944

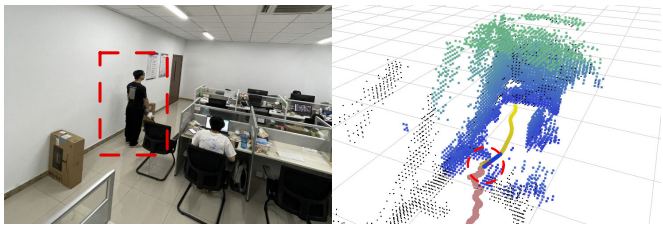

Fig. 1. The Smart Cane is an RNA-based device. It incorporates perception, localization, planning, and control functions, which could offer safe and effective guidance for individuals with visual impairments. Real-world demonstrations have effectively showcased the use of the Smart Cane for navigation.

configurations for this device, and the design of the hardware structure is described in Section III-A.

The primary objective of the Smart Cane is to address the key challenges faced by individuals with visual impairments in navigating indoor environments. Leveraging the capabilities of the Smart Cane, we have developed a comprehensive navigation framework that incorporates localization, perception, planning, and control. This software framework is described in Section III-B.

This framework utilizes advanced Visual-Inertial odometry [10] or, alternatively LIDAR-Inertial Odometry [11] to provide localization for both the system and individuals with visual impairments, addressing the absence of GPS positioning in indoor environments. Additionally, there are many static obstacles in the real environment, i.e., hanging obstacles, etc. This allows us to construct an ESDF (Euclidean Signed Distance Field) [12] map, which is subsequently supplied to the planning and control modules. Furthermore, we have developed a walking path generation algorithm, specifically customized to assist visually impaired individuals in indoor environments characterized by a higher obstacle density.

Overall, the contributions of this study can be summarized as follows:

(1) A safe and efficient walking path generation method was developed. This method combines LIPM (Linear Inverted Pendulum Model) and LFPC (Linear Foot Placement Controller) motion primitives to generate walking paths for the visually impaired. (see Section IV).

(2) The proposed algorithm was implemented on the Smart Cane, followed by analysis and verification through indoor experiments. Besides, comparative experiments in simulation environments were also conducted (see Section V).

II. Related Work

A. Navigation System for Visually Impaired

Research and commercial efforts have focused on developing real-time navigation systems for individuals who are limited vision or visually impaired. These systems can be classified into four types: mobile platforms, guide dog robots, wearable devices, and robotic canes. Certain works, such as mobile platforms [3] and guide dog robots [13], encountered limitations in mobility due to the substantial size and weight of the robot within confined indoor spaces. Furthermore, these studies did not take into account characteristics of human walking, which could potentially lead to user discomfort. Moreover, the wearable device lacks intuitive steering assistance, as mentioned in the study by Katzschmann et al. [4].

Recently, there has been a growing emphasis on developing robotic canes [3], [5], [14], [15]. It is a compact device that utilizes machine vision and robotics technology to adapt to challenging environments. However, there is a lack of intuitive steering assistance in Co-Robotic Cane [5] for visually impaired individuals. To address these limitations, the Augmented Cane [14] was developed with the capability of environment sensing and intuitive steering assistance. However, due to the inherent nature of 2D radar and RGB camera sensors, the device lacks the capability to map three-dimensional spaces, leading to limitations in its capacity to plan routes effectively and provide comprehensive guidance to users.

Building upon these studies, autonomous framework technology is adopted in our Smart Cane to address the navigation challenges for individuals with visual impairments in complex indoor environments.

B. Walking Path Generation

1) Path Generation: Path generation is a well-established research area in robotics, particularly in scenarios where feasible paths need to be generated while considering kinodynamic constraints. Kinodynamic planning, which was introduced by Donald et al. in 1993 [16], considers both the kinematics and dynamics of a system to generate paths or trajectories that adhere to these constraints.

Hybrid A* [17], [18] is a kinodynamic planning algorithm specifically developed for autonomous driving vehicles, combines both discrete and continuous state-space search techniques to efficiently generate feasible paths for autonomous vehicles. In the context of UAVs (Unmanned Aerial Vehicles), the Fast-planner introduced by Zhou et al. [19] utilizes a state space model of UAVs to expand motion primitives outward, resembling the search process of A*. Falco is another local planner [20], which generates cubic spline curves as a set of feasible paths offline.

The aforementioned algorithms highlight the significance of taking kinodynamic constraints into consideration and utilizing search techniques to generate viable paths in various domains, including autonomous driving and UAVs. Inspired by these algorithms, we tailored our method specifically to address the requirements of navigation for visually impaired individuals.

2) Human Walking Model: Our proposed walking path generation method is based on the principles of the human walking model. Previous research have focused on two key aspects: accurately representing real walking trajectories using mathematical models [21], [22], [23] and implementing gait control strategies on biped robots [24], [25], [26], [27]. These efforts have significantly contributed to our understanding of human locomotion and the advancement of walking-related technologies.

Papadopoulos et al. [21] used motion capture to collect human paths, and used inverse optimal control to obtain a cost function to generate the human walking paths. However,

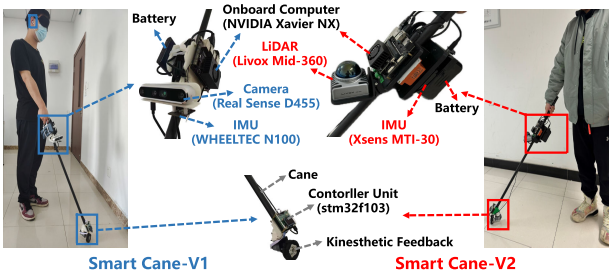

Fig. 2. The hardware componences of this Smart Cane. In the figure, blue elements denote v1-specific hardware, red represents v2-exclusive components, and black indicates shared hardware between both versions.

their unicycle kinematic model lacks gait points and overlooks collision detection.

The development of bipedal robots has prompted extensive research on the challenges of walking dynamics and balance control. The conventional gait generation approach, based on the ZMP (Zero Moment Point) [24] concept. The elongated number of iterations in the solving process leads to a slower solving speed. Furthermore, Kajita et al. [26] introduced the SQD (Spatially Quantized Dynamics) method. However, the SQD-based approach requires offline gait generation, which limits its applicability to high-level tasks like path planning. The LFPC (Linear Foot Placement Controller) [25] utilizes a linear function to generate stable gaits with faster convergence compared to the ZMP-based approach. In the field of path planning, planners designed based on LFPC can result in reduced computational overhead. Extensive studies have demonstrated the effectiveness of LFPC in various gait generation applications, positioning it as a promising solution in the field.

Therefore, we propose a new method for generating walking paths that combined LIPM and LFPC. This approach is specifically designed to assist individuals with visual impairments in navigating their surroundings.

III. System Overview

A. Hardware Architecture

Two configurations of the Smart Cane have been designed, utilizing the following hardware components, with a combined weight of 1.5kg. Figure 2 illustrates the distinct subsystems of both Smart Cane versions (V1 and V2), showcasing the arrangement and interconnection of sensors, high-level navigation unit, low-level control unit, and battery.

1) Power: We use a 10,000 mAh LiPo batteries with an output capacity of 22.5W to power our system. The NVIDIA Xavier NX requires 10W. In the V1 version, both the Realsense D455 and IMU modules require 5W each. In the V2 version, it involves a multi-line LiDAR and IMU modules, both consuming 5W of power. Besides, kinesthetic feedback component requires a power of 5W. According to these power requirements, the selected batteries could meet the power demands of our platform. The batteries exhibit a capacity to sustain the complete operation of the system for a duration spanning between 4 to 5 hours.

2) High-Level Navigation Unit: To ensure optimal computing performance for the high-level navigation unit in the Smart Cane, we have selected the NVIDIA Xavier NX as the platform. The Xavier NX offers a six-core CPU, a 384-core GPU, and 8GB of RAM, enabling efficient execution of localization, planning, high-level control, and other task-specific algorithms. This hardware configuration ensures real-time and accurate processing for navigation tasks.

3) Low-Level Control Unit: To provide steering assistance for visually impaired individuals, we have incorporated a kinesthetic feedback into the Smart Cane. This feedback is compored of an omnidirectional wheel and a motor [14]. To control the torque output of the motor, we utilize an STM32F103 MCU (Microcontroller Unit) in our platform. The MCU establishes communication with the high-level navigation unit of the Smart Cane via a serial port, facilitating synchronized control and seamless integration of the kinesthetic feedback with other navigation features. By controlling the motor's rotation, the input from the controller is converted into directional steering.

4) Sensors: The Smart Cane in its v1 iteration incorporates an Intel Realsense D455 camera for capturing grayscale and depth images, enabling mapping and localization. Furthermore, an IMU (Inertial Measurement Unit) is utilized to collect orientation and motion data. These sensors play a crucial role in environment perception and accurate navigation.

In the V2 version, the Smart Cane integrates a multi-line LiDAR that provides 3D point cloud information along with an IMU for gathering orientation and motion data. This combination of sensors ensures the acquisition of detailed 3D environmental data and accurate motion information, enabling enhanced environmental perception and precise navigation for the Smart Cane.

Additionally, we have integrated Bluetooth headphone into our system to facilitate voice feedback. This allows real-time communication of important information and instructions to visually impaired individuals, enhancing their navigation experience and providing essential guidance and updates.

B. Software Architecture

The software architecture of the Smart Cane is designed as a comprehensive system solution for limited vision navigation, incorporating perception, localization, planning and control functions. All these functions are implemented on the microcomputer of the Smart Cane, providing a compact and efficient solution. To ensure safe navigation in complex environments, we have developed an pipeline framework that integrates complete perception, localization, planning and control functionalities. Figure 3 illustrates the overall software framework, demonstrating the seamless integration of these components for effective and reliable navigation assistance.

1) Localization: To accurately estimate the real-time position of the Smart Cane in its v1 version, we utilize VINS (Visual Inertial Navigation System) as a visual odometry framework, as described in [10]. VINS is a robust VIO (Visual Inertial Odometry) system that combines visual and inertial sensor measurements to estimate the motion and pose of the Smart Cane.

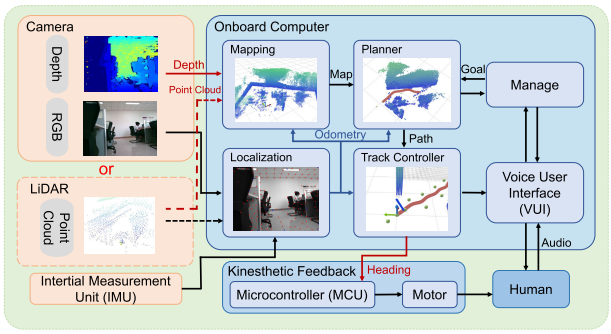

Fig. 3. This is the system architecture of Smart Cane, which enables the implementation of localization, mapping, planner, and track controller functionalities in a locally online manner. The primary objective of this architecture is to ensure the safe guidance of individuals with visual impairments.

In the V2 version, we leverage a faster-LIO [11], a robust LiDAR-Inertial Odometry (LIO) algorithm, to achieve accurate real-time positioning. This algorithm efficiently combines data from the LiDAR and IMU sensors to estimate the Smart Cane's motion and pose.

The improved localization data obtained from the VINS algorithm (in the V1 version) or faster-LIO (in the V2 version) is subsequently employed by different modules within the system. This enables the Smart Cane to accurately estimate its position and orientation in real time, facilitating effective navigation.

2) Mapping: In the V1 version, to represent the environment, we employ a fusion approach that combines depth measurements from the stereo camera with pose estimation. In the V2 version, we extend the environment representation by incorporating 3D point cloud data provided by the multi-line LiDAR.

This enables the mapping of an EDF (Euclidean Distance Field) voxel map of the environment [12]. The map is updated at a frequency of 10 Hz using an efficient O(n) algorithm. Additionally, we adopt an incremental update strategy to efficiently update the voxel grid within the sensing range [19]. This approach ensures real-time performance and efficient map maintenance.

3) Planner: In this article, we propose the new walking path generation method. This method combines LIPM and LFPC motion primitives to generate walking paths for visually impaired individuals. The algorithm includes a dynamic path search that prioritizes safety throughout the process.

4) Track Controller: To guide the heading angle of visually impaired individuals, we implement ground kinesthetic feedback based on the work of Slade et al. [14]. The output motor's motion is tracked and controlled by the controller, which performs feedback control by comparing the current pose information with the expected output information. This facilitates effective guidance of visually impaired individuals along the planned path generated by the planning module.

Our system, with its adaptable and modular structure, provides the flexibility to incorporate hardware components from both the Smart Cane v1 and v2 versions. This adaptability is instrumental in conducting thorough testing and validation of planning algorithms across simulated and real-world settings.

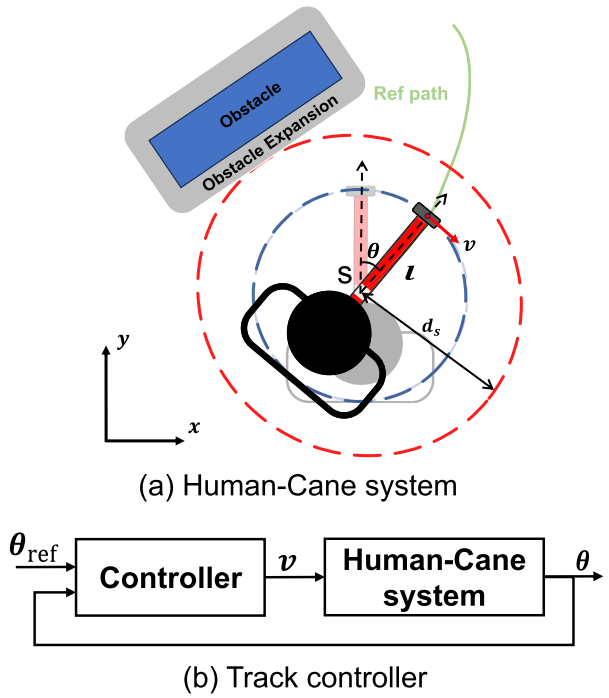

Fig. 4. (a) Illustrates a 2D schematic of the human-cane system. Here, the combination of the individual and the smart cane is simplified into a system with a state represented as (x, y, θ). (b) demonstrates the closed-loop structure of the track controller. Given a reference θ, the controller outputs the motor velocity, influencing the θ of the human-cane system.

By harnessing the capabilities of our Smart Cane hardware, we can conduct comprehensive assessments of algorithm performance, ensuring robustness and efficacy across diverse environmental scenarios and conditions.

C. Human-Cane System Model

To illustrate how the SmartCane system effectively guides the visually impaired, we simplified the relationship between individuals and the SmartCane. The participants were instructed to use the SmartCane before the experiment.

For ease of analysis, we considered the individual and the cane as a unified system termed the 'Human-Cane System', as shown in Figure 4(a), analyzed on a 2D plane. We define system state as $x = (x, y, \theta) \in \mathbb{R}^3$ in the world frame, where x and y represent the position of the system's position, and θ represents the heading angle of the system. d_s represents the design of a safety distance, ensuring safe obstacle avoidance for the human body. Additionally, l stands for the distance from the front end of the cane to the Kinesthetic Feedback, represented as the projected distance on the 2D plane, depicted by the blue circular motion in the diagram. On the other hand, v denotes the linear velocity generated by the Kinesthetic Feedback motor. The system adjusts θ based on changes in v, whereas the actual positional movement relies on the individual's walking.

Our system operates by utilizing a planner algorithm to chart a reference path. This reference path is utilized to guide the "Human-Cane System", allowing it to navigate around obstacles in the environment by controlling θ through the Kinesthetic Feedback.

D. Kinesthetic Feedback Track Controller

For track controller, a basic Proportional-Integral (PI) type controller is employed. Furthermore, in Figure 4 (b), the controller uses the reference heading provided by the Planner to modify and control the speed output to the motor. Thus, the following form of the PI controller is adopted.

$$v = K_p\,(\theta_{\text{ref}} - \theta) + K_i \int_0^t (\theta_{\text{ref}} - \theta) d\tau \tag{1}$$

where v is the non-dimensional speed control signal, K_p and K_i are the proportional and integral gain respectively.

The controller coefficients K_p and K_i were established through extensive testing, following a robust control parameter space approach [28]. Additionally, we conduct a simple experiment to adjust K_p and K_i, which enables the device's output to influence the user's walking pattern.

IV. Walking Path Generation

In this section, we provide a detailed explanation of our proposed method. Drawing inspiration from the principles of hybrid A* [18], we have tailored our approach to specifically address the requirements of navigation for visually impaired individuals. Our method is specifically designed to operate in real time on small, resource-limited platforms.

A. Problem Definition

The problem of generating walking paths for visually impaired individuals is a crucial task that involves ensuring both safety and efficiency. It can be formulated as a search problem, taking into account several inputs. These inputs include an environment map represented as an ESDF (Euclidean Signed Distance Field) map, the initial human-cane system state denoted as $s_o = (x, y, \theta)_o$, and the target state of human-cane system denoted as $s_g = (x, y, \theta)_g$.

The planner is designed to follow the human dynamic constraints. Human dynamic constraints refer to the kinematic limitations experienced during human locomotion. For example, when a person is walking at a certain speed, they are unable to make rapid turns. If the planner generates excessive turning commands during the walking process, it may result in significant deviations in the guidance system, potentially leading to a sharp turn. In extreme cases, this could cause the user to fall.

The objective is to find a path that not only adheres to human dynamic constraints but also guarantees mobility and comfort for the visually impaired individual. The path should effectively guide the limited vision person from the initial state to the target state. To achieve this, the Smart Cane's grounded kinesthetic feedback tracking control is employed.

B. Walking Path Search

Our walking path search method leverages the LIPM (Linear Inverted Pendulum Model) [24] as a continuous motion model and the LFPC (Linear Foot Placement Controller) [25] to generate future walking points. By integrating these components, we can ensure that the generated walking path satisfies human kinematic constraints, promoting mobility and comfort for visually impaired individuals.

Algorithm 1 Walking Path Searching

```
Initialize Search
Initialize LFPC
while O is not empty do
  n_c ← O.pop()
  C.insert(n_c)
  if NearEnd(n_c) and ReachHorizon(n_c) then
    return RetrievePath(n_c)
  end if
  inputs ← LFPC.setCtrlParam()
  for each um in inputs do
    LFPC.updateOneStep(um)
    n_p ← LFPC.getNodeState(um)
    Prune(n_p)
    if checkSafe(n_p) and C.contains(n_p) then
      g_temp ← n_c.g_c + Cost(n_p)
      if O.contains(n_p) then
        O.add(n_p)
      else
        if g_temp ≥ n_p.g_c then
          continue
        end if
      end if
      n_p.parent ← n_c
      n_p.f_c ← g_temp + Heuristic(n_p)
    end if
  end for
end while
```

To accomplish this objective, we employ the A* algorithm to systematically search for a safe and optimal path that aligns with the specified constraints. Algorithm 1 outlines the search process employed in our method, which shares similarities with the A* algorithm. The open set **O** and closed set **C** in our approach correspond to their counterparts in the A* algorithm. However, there are some variations that distinguish our method.

The data structure N stores expanded primitives, representing nodes or elements derived from the initial primitives in the search space. N likely functions as a container to manage these elements during the search.

To optimize our search process, we use the **Prune()** function, eliminating nodes within the same grid as the one with the minimum gc value. Here, gc stands for the cost accumulated from the start to the current node along the path. This step reduces unnecessary exploration.

After pruning, the **CheckSafe()** function ensures the safety of remaining nodes, considering specific constraints. It verifies if nodes meet safety criteria, such as collision avoidance or other domain-specific constraints.

During the algorithm, if a node n_c popped from set O isn't close to the target and reaches the exploration boundary, we update a new node n_p using **updateOneStep()**. The algorithm then calculates the gc value for n_p, representing the cost along the discovered path.

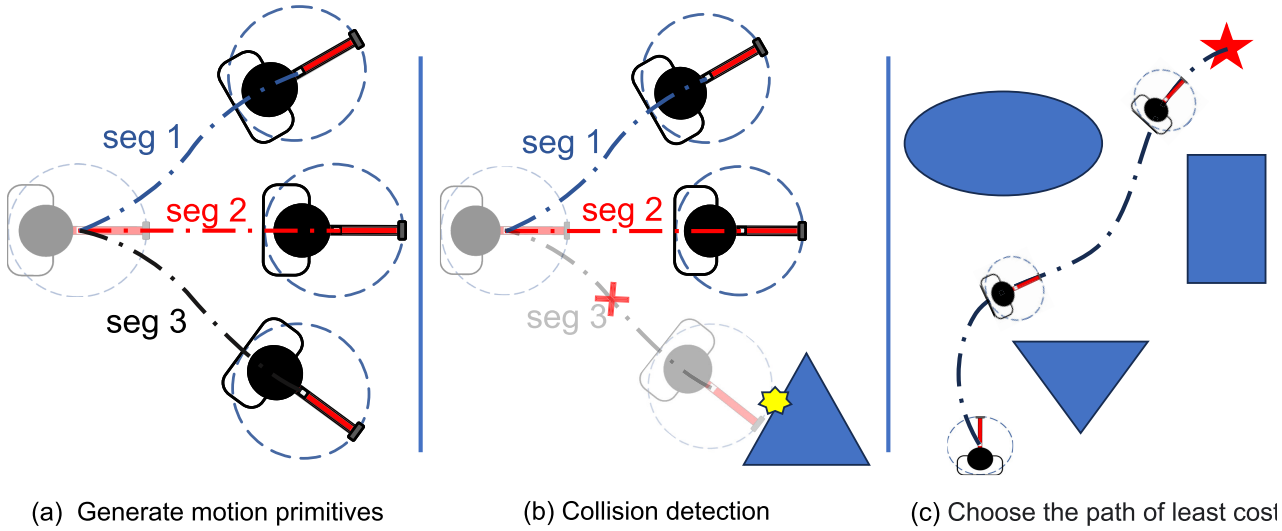

Fig. 5. This is the "Walking Path Generation" process. In each search cycle, we follow the steps shown in (a)-(c). In step (a), we generate motion segments using LFPC and LIPM. In step (b), these generated segments undergo collision checking, removing any segments that collide with the environment. Finally, in step (c), the path with the lowest cost is chosen and output.

Following safety checks, the algorithm finds the minimum cost and gc value path within the map. This involves exploring paths to identify the safest and most cost-efficient path within specified constraints.

In our methodology, we employ specific LFPC parameters to generate various candidate gait points. These gait points act as inputs for the LIPM, enabling the creation of motion primitive segments—referred to as the Center of Mass (CoM) path within the ESDF map for expansion. This process aligns with the "Walking Path Generation" approach depicted in Figure 5. Subsequently, these generated motion segments undergo collision checking to refine the path, ensuring avoidance of collisions with the surrounding environment. Ultimately, the selection of the path with the lowest cost becomes the final output, showcasing our method's optimization for both efficiency and collision-free traversal.

C. LIPM and LFPC

The algorithm includes an important function called **updateOneStep()**, which leverages the LFPC and LIPM to enable forward expansion [25]. By utilizing LFPC and LIPM, we can generate motion primitives (CoM, Center of Mass path) within the voxel map, facilitating the search for a path that guides the user from the starting state to the end state.

1) LIPM: The LIPM (Linear Inverted Pendulum Model), as depicted in Figure 6, simplifies the walking module to a point mass at a fixed height with adjustable leg lengths [24]. This simplified representation allows for the analysis and design of control strategies for walking motion.

In the continuous phase step, the LIPM is described by a set of differential equation:

$$\ddot{x} = \frac{gx}{h} \tag{2}$$

Equation (1) can be solved to obtain the position and velocity of the point mass, which are:

$$\begin{aligned} x(t) &= x(0)\cosh(t/T_c) + T_c\dot{x}(0)\sinh(t/T_c) \\ \dot{x}(t) &= x(0)\sinh(t/T_c) + T_c\dot{x}(0)\cosh(t/T_c) \end{aligned} \tag{3}$$

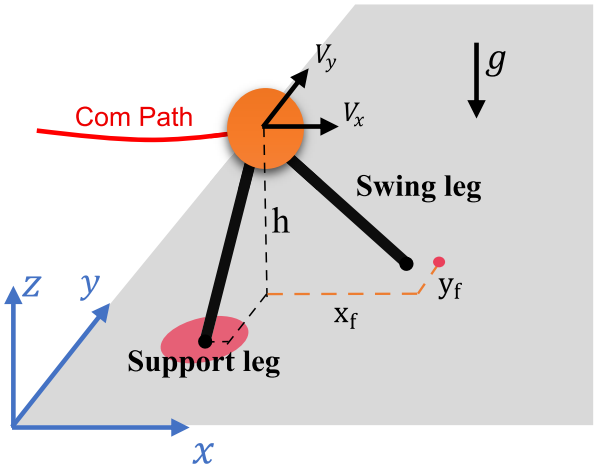

Fig. 6. An illustration of the generation of gait points and CoM paths using LIPM and LFPC. LFPC determines the future gait point, while LIPM solves for the corresponding CoM path.

Here, h is the height of the CoM, g is the acceleration due to gravity, and $T_c = \sqrt{h/g}$.

The LIPM can be extended to the three-dimensional case by decoupling the two axes and applying the same principles, as follows:

$$\begin{aligned} \ddot{x} &= \frac{gx}{h} \\ \ddot{y} &= \frac{gy}{h} \end{aligned} \tag{4}$$

2) LFPC: The LFPC (Linear Foot Placement Controller), as shown in Figure 6, proposed by Ye et al. [25], calculates the subsequent body position based on predefined controller parameters and the body velocity at the moment of foot contact. It predicts the foot landing position using a linear function that depends on the body velocity, which is:

$$x_f = a + bv \tag{5}$$

x_f is the next step position relative to CoM (CoM as the origin), v is the body velocity at the moment of falling, a and b are the controller parameters.

Based on the given formula and information, the foot landing points in the x-axis and y-axis directions can be designed as follows:

$$\begin{aligned} x_{fl} &= a_w + bv_x \\ x_{fr} &= a_w + bv_x \\ y_{fl} &= a_l + bv_y \\ y_{fr} &= -a_l + bv_y \end{aligned} \tag{6}$$

where x_{fr}, x_{fl} represent the placement positions of the two legs in the x-axis direction, y_{fr}, y_{fl} represent the placement positions of the two legs in the y-axis direction, and a_w, a_l, b are controller parameters.

To incorporate the walking direction angle and describe the walking gait pattern, a triplet (d_l, d_w, θ) is used, where d_l represents the step length, d_w represents the step width, and θ represents the angle between the walking direction and the +x direction.

The LFPC allows for convenient adjustment of step length, step width, and walking direction in 3D walking. We can obtain a transformed foot landing position, which are:

$$\begin{aligned} x_{fr} &= -a_l \times \cos(\theta) + a_w \times \sin(\theta) + b \times v_x \\ y_{fr} &= -a_l \times \sin(\theta) - a_w \times \cos(\theta) + b \times v_y \\ x_{fl} &= -a_l \times \cos(\theta) - a_w \times \sin(\theta) + b \times v_x \\ y_{fl} &= -a_l \times \sin(\theta) + a_w \times \cos(\theta) + b \times v_y \end{aligned} \tag{7}$$

In the cited work [25], the Poincaré map was employed to derive the equilibrium criterion of the LFPC. The study showed that when the stable coefficient b of the LFPC velocity feedback falls within a specific range, it satisfies the stability condition of this control law. As shown in the following formula:

$$T_c \frac{c_T - 1}{s_T} < b < T_c \frac{c_T + 1}{s_T} \tag{8}$$

3) Analytic Expansions: The LFPC plays a crucial role in generating multiple gait points by adjusting the parameters (a_l, a_w, θ), which determine the characteristics of foot landing positions in terms of step length, step width, and walking direction. A walk simulation is shown in Figure S1. This enables the generation of diverse walking gaits tailored to the specific requirements of visually impaired individuals.

To ensure a continuous and realistic generation of motion paths, the LIPM is utilized. The LIPM simplifies the complex dynamics of human walking by representing the walking system as a point mass at a fixed height. This simplified model allows for efficient analysis and simulation of human walking motion.

During the search process for a suitable path, different combinations of LFPC input parameters are simulated within the LIPM for a single gait cycle. Each simulation generates a unique gait pattern, represented by continuous motion primitives. These motion primitives are then expanded and incorporated into the search tree as part of the path planning process.

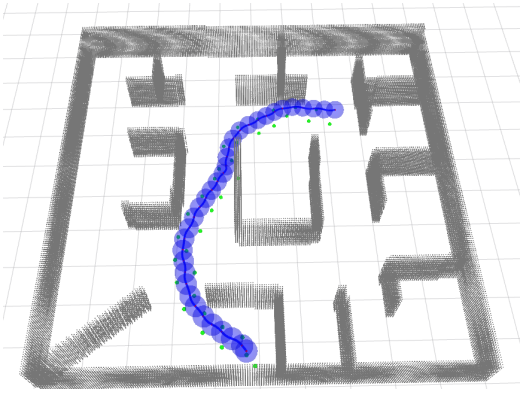

Fig. 7. In the simulation environment, the collision detection algorithm is visualized in a top view. The visualization includes a blue sphere representing the detection range with the CoM point as the center, and a green sphere representing the detection range with the gait point as the center.

By employing this expansion method, the planned path considers the kinematic constraints related to the human body within the human-cane system assumption. This assumption aligns the smart cane's movement speed with that of an individual. Our algorithm relies on a state estimation specific to the human-cane system. It is essential to note that our algorithm aims to generate a reasonable path rather than an optimal one, as it doesn't precisely calculate the body speed needed. This approach ensures that the planned path respects the natural motion capabilities and limitations of visually impaired individuals, and promotes safe and comfortable navigation.

D. Collision Detection

To ensure the safety of the generated path, the algorithm incorporates collision detection between the path and the voxel grid map. This collision detection is performed using the **CheckSafe()** function in Algorithm 1. The planner simultaneously plans the CoM path and future gait points for the visually impaired individual. The gait points guide the walking direction but do not directly control the actual landing points.

For the purpose of collision detection, a simplified approximation of the human lower limbs is utilized. This representation consists of three spheres with different radii, which approximate the shape of the lower limbs. These spheres are centered at the current center of mass point, the previous gait point (support leg point), and the upcoming gait point, respectively. Figure 7 illustrates a top-down view of these three spheres for collision detection. The blue sphere represents the collision detection range centered around the center of mass point, while the green sphere represents the collision detection range centered around the gait point.

Collision detection plays a critical role in the context of online motion planning. The ESDF map [12] is an essential tool for collision detection. The ESDF map provides valuable information regarding obstacle distances and gradients, which enables efficient collision detection. By leveraging the current ESDF voxel index, it becomes possible to directly compute the distance from a point to an obstacle within the ESDF map, resulting in low computational costs. This direct calculation approach effectively transforms the physical 3D environment

into a 2D representation, facilitating accurate collision detection and supporting the generation of paths in real-time.

E. Motion Primitives Cost

In the process of assisting limited vision individuals with a Smart Cane, it is desirable to minimize the curvature of the planned CoM path. This helps reduce torque between the Smart Cane and the limited vision individuals during turning, preventing overloading of the Smart Cane's motor and maintaining the limited vision's balance.

To account for movement restrictions and the comfort of visually impaired individuals, a penalty function method is employed during heuristic numerical calculation and sub-node expansion. For each path generated by the LFPC, the cost function for the path between nodes n_i and n_{i+1} can be defined as:

$$c(n_i, n_{i+1}) = D(n_i, n_{i+1}) + \sigma \times \theta + risk_{i+1} \tag{9}$$

$$risk_{i+1} = \begin{cases} w_o \cdot (d_{max} - d_{obstacle}) \\ 0 \end{cases} \tag{10}$$

Here, $D(n_i, n_{i+1})$ represents the Euclidean distance between the two nodes. σ is the penalty coefficient for generating turns, which influences the cost based on the angle θ between the path segments. Additionally, $risk_{i+1}$ represents the safety risk penalty value associated with the next node.

By summing these three components, the cost function accounts for the distance traveled, the influence of turns, and the potential risk due to obstacles or other safety considerations. This allows for the evaluation and comparison of different path options during the planning process.

According to the provided definition, in Algorithm 1, the calculation of the **Cost()** function in the A* algorithm involves the cost of the motion primitives generated by discretized input.

V. Experiment and Analytic

The experiment aims to validate the effectiveness of the Walking Path Generation (WPG) algorithm and its real-world applicability for guiding users. Three experiments were designed:

(1) System-Level Testing: Comparison between smart cane and white cane in normal subjects who are blindfolded. Evaluate the navigational performance of the Smart Cane utilizing WPG against White Cane in unfamiliar indoor environments to assess the practicality and usability of the new system.

(2) WPG Performance Analysis: Simulated comparison of WPG and the A* algorithm. Compare the performance differences between WPG and the A* algorithm in simulated environments to highlight the advantages of the WPG method.

(3) Guidance Experiments Evaluation: Comparative analysis of multiple methods in real environments with visually impaired subjects. Execute comparative experiments utilizing diverse methods in real-world settings to highlight and emphasize the primary benefits offered by the WPG approach.

These experiments provide empirical evidence and insights into the proposed method's effectiveness and usability, both in simulated and real-world scenarios.

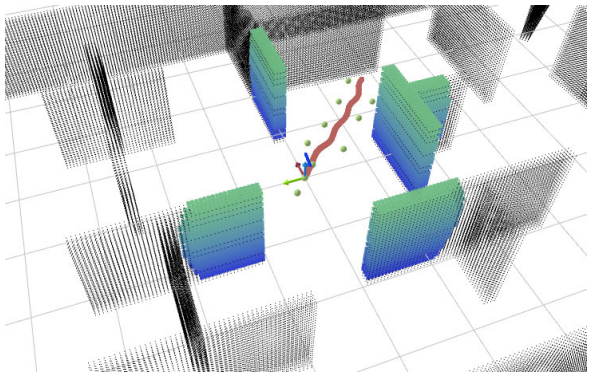

Fig. 8. The Re-Planning strategy focuses on planning path within the perceived area. In the figure, the white dots represent the unknown area where information is not available, while the colored dots represent the perceived area where sensory data is obtained.

A. Experiment Settings

The experiment settings of the experiment are as follows:

1) : The walking path planning method and the overall navigation system software described in this paper were implemented using the Robot Operating System (ROS) framework. All experiments were conducted using an onboard computer and a device designed for the Smart Cane. We set $g = 10\ m/s^2$, $b = 0.4$, $0.4 \leq a_l \leq 0.6$, $0.1 \leq a_w \leq 0.2$, and $0° \leq \Delta\theta \leq 30°$ for LFPC and LIPM to generate motion primitives. The duration of each step of LIPM is set to $t_{sup} = 0.3s$.

2) : Our approach utilizes a Re-Planning strategy for dynamic path planning. It is important to note that planning a path in an unknown environment may not yield meaningful results. Thus, we address this challenge by updating the map using sensor sensing range increments and generating a path within that range. Our goal is to guide limited vision individuals as close to their destination as possible. In Figure 8, we present a demonstration of our Re-Planning strategy in a simulation environment.

3) : Our experiment utilizes a kinesthetic feedback track controller, employing the Proportional-Integral (PI) control methodology (refer to Section III-D). Preceding the formal experiment, the controller parameters of K_I and K_P were tuned specifically. The K_I and K_P settings vary among individuals but adhere to the parameter ranges outlined as follows: $0.3 \leq K_p \leq 0.5$, $0.02 \leq K_I \leq 0.04$.

4) : The experiment protocol was approved by the Nanjing Brain Hospital Institutional Review Board (2023-KY116-01). All participants were volunteers who were provided written informed consent before the experiment.

B. System-Level Testing

The experiments were conducted to compare the performance of users navigating with a white cane versus using the Smart Cane in key activities. All ten participants received detailed information about the study's objectives and experimental procedures. They were blindfolded individuals without visual impairment and lacked prior experience with white canes. Before the experiments commenced, each participant underwent a brief tutorial on operating the Smart Cane.

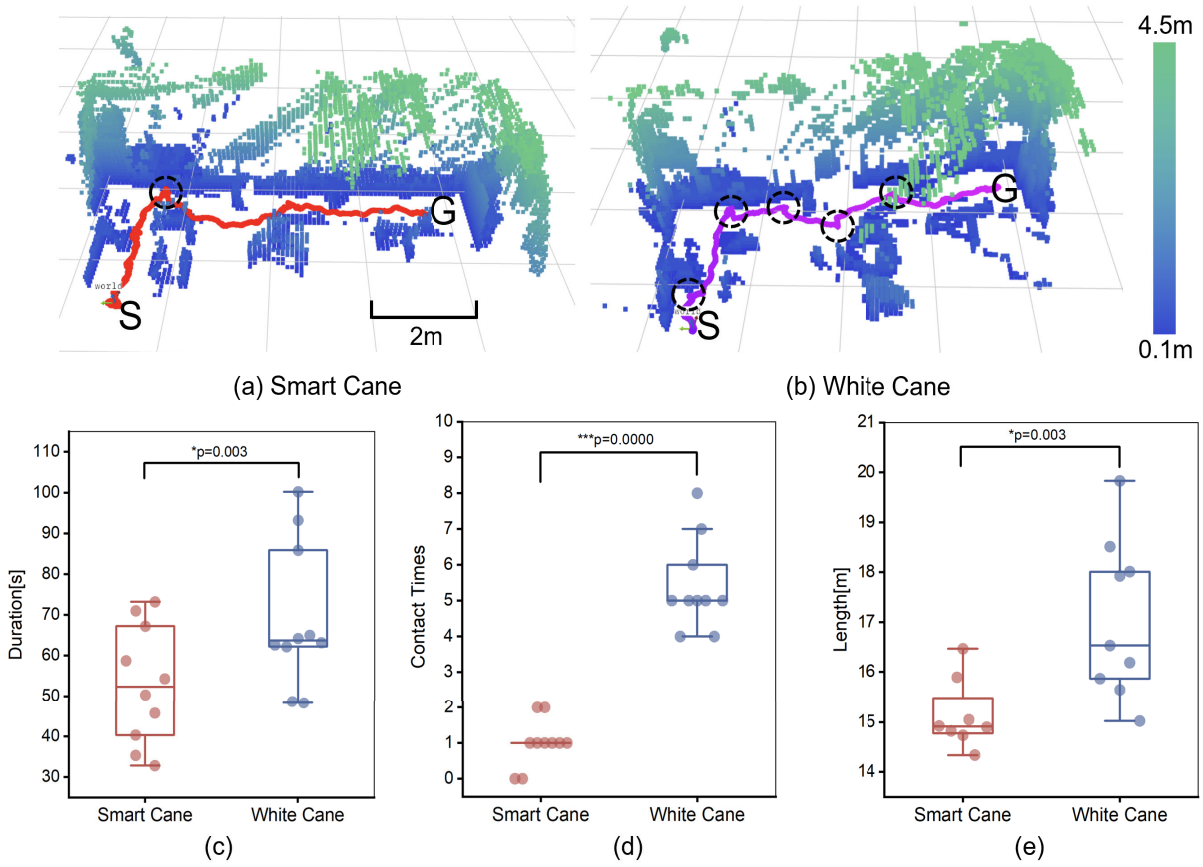

Fig. 9. The navigation performance of Smart Cane versus White Cane. For a representative participant, Smart Cane is used in (a) while white cane is used in (b) (S:start point; G:goal point; The black circles are the location of the collision with the obstacle). The duration time of two canes is shown in (c). The contact times of two canes are shown in (d). The length of paths of two canes is shown in (e). A two-way ANOVA model was used to analyze the significance of the effect of using the Smart Cane or a white cane on the participants. ($*p < \mathbf{0.05}$,$**p < \mathbf{5\times10^{-3}}$, $and***p < \mathbf{5\times10^{-4}}$).

None of the participants had prior experience using guide canes. Prior to the experiments, the walking planner and feedback intensity were individually adjusted based on each participant's walking habits. The relevant information can be found in Table S1 for reference. Throughout the experiments, participants were instructed to navigate a pre-defined path as accurately and swiftly as possible. The path layout included randomly positioned obstacles marked as 'a,' 'b,' 'c,' and 'd,' as illustrated in Figure S2.

In the experiment, the participants randomly used the guide cane or the white cane first. The evaluation metrics used in the experiments included the duration time of tasks, the number of contact times with obstacles, and the length of the path. The length of the path was measured using the VINS integrated into the Smart Cane for each participant.

The navigation performance of Smart Cane versus White Cane are depicted in Figure 9. In Figures 9(a) and (b), we present the results of a representative participant utilizing either the Smart Cane or White Cane for navigation. The participant's walking trajectory exhibits smoothness and consistently maintains a safe distance from environmental obstacles. Notably, it can be observed that there is one contact with obstacle in Figure 9(a) and five contacts in Figure 9(b). This result supports the effectiveness of the Smart Cane in assisting the participant to avoid collisions and navigate along a safe path.

The statistical results of duration time, contact times and length of path are shown in Figure 9(c)-(e), respectively. A two-way ANOVA (analysis of variance) was used for statistical analysis. Notably, all participants finished the experiment. Using the Smart Cane, subjects took an average of 52 seconds to walk through the corridor, compared to 70 seconds with a white cane. Smart Cane users experienced only 1 collision, while white cane users had an average of 6 contacts. On average, Smart Cane users walked 15.05m, while white cane users covered 16.71m. The box plot demonstrates concentrated distance values for Smart Cane users while significant individual variation with the white cane. These findings underscore the advantages of the Smart Cane over traditional white canes, including shorter duration, fewer contacts/collisions, and more consistent path lengths.

C. WPG Performance Analysis

A* algorithm was also implemented for comparing with our Walking Path Generation (WPG) algorithm [29]. We conducted experiments in four distinct environments: "office1", "office2", "pillar" and "school indoor". The "school indoor" map utilized in our experiments was obtained from the publication by [30]. This map (Figure 10 (a)) comprises long and

TABLE I
PATH GENERATION COMPARISON

Map	Resolution(m)	Size(m)		Times(s)		Use_nodes		len(m)	
				A*	Proposed	A*	Proposed	A*	Proposed
office1	0.1	40 × 40 × 4.5	Avg	0.012	0.007	13290.87	1198.96	29.38	28.00
			Std	0.006	0.006	5839.07	834.49	4.92	4.25
			Max	0.024	0.020	22074.00	2893.00	20.54	20.44
office2	0.1	40 × 40 × 4.5	Avg	0.022	0.012	21974.57	1635.89	39.93	38.47
			Std	0.010	0.010	8963.42	1214.35	7.84	7.02
			Max	0.043	0.043	35475.00	4885.00	24.28	24.41
pillar	0.1	40 × 40 × 4.5	Avg	0.045	0.002	39533.79	431.20	35.92	32.68
			Std	0.027	0.001	21343.39	169.89	7.36	5.96
			Max	0.100	0.005	84869.00	985.00	24.84	24.55
school indoor	0.2	250 × 150×5	Avg	0.042	0.423	28325.57	35547.48	224.06	227.85
			Std	0.019	0.307	10306.32	26313.27	44.63	45.50
			Max	0.074	1.076	40866.00	90590.00	139.14	140.51

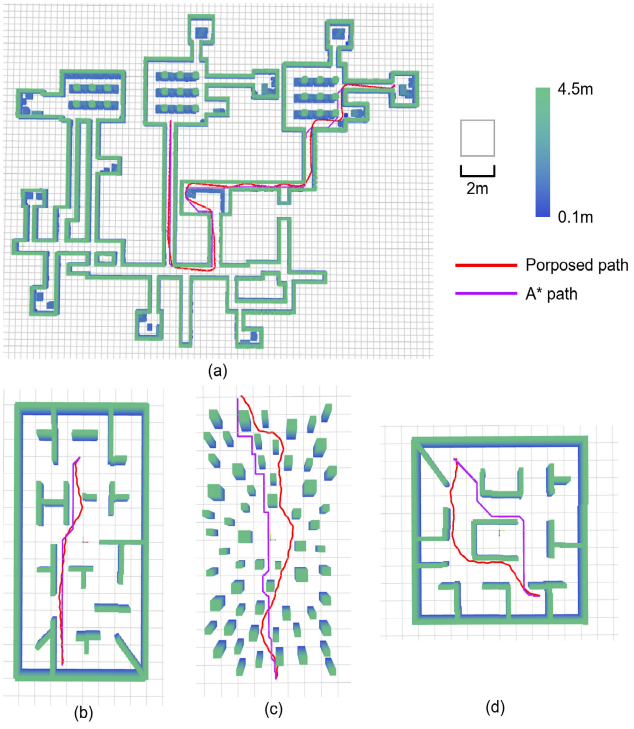

Fig. 10. Generate paths in four different simulation maps and compare them to evaluate their performance: (a) school indoor (b) pillar (c) office1 (d) office2.

narrow corridors interconnected with lobby areas and includes various obstacles such as tables and columns. Additionally, the "office1", "office2" and "pillar" maps were obtained from [19]. These maps encompass various spatial layouts, including office environments and specific structures such as pillars. These maps were selected due to their representation of typical features and characteristics found in various real-world environments.

To ensure a fair comparison, we conducted 30 runs of the algorithm in each environment. The comparison of the paths generated in the four maps is shown in Figure 10. The evaluation metrics included the running time of the search (time complexity), the number of nodes visited (space complexity), and the length of the final generated path (same starting and ending points). The results of these experiments are summarized in Table I.

Based on the statistical results, our method outperforms A* in environments where small-scale planning is required (less than 100m, in scenarios of office1, office2, pillar). It demonstrates less execution time and storage space. Additionally, the path generated by our method is represented in (x, y, θ) space, whereas A* operates in (x, y) space.

One of the key advantages of our algorithm is the utilization of expanding motion primitives for search. In contrast, A* expands its search based on the map resolution. As a result, our method significantly reduces the number of iterations for searching when planning on a small-scale map. It should be noted that when generating large-scale paths (greater than 100m, in scenarios of school indoor), the search efficiency of our method is reduced. This is because each search in our method requires generating motion primitives, which can be time-consuming.

Since the Re-Planning strategy is adopted in Smart Cane, we focus on generating paths within the sensing range of the sensor. Typically, this range is small-scale (in our implementation, 10m). In this small-scale planning environment, our method can generate plans within tens of milliseconds, enabling us to achieve a replanning frequency of 10Hz. This allows for real-time adaptability and efficient navigation in dynamic environments.

D. Guidance Experiments Evaluation

This study aimed to evaluate navigation methods in real-world scenarios with visual impaired participants. The information of these participants can be found in Table S2 and Table S3 of the supplementary materials. They also had 3 to 5 years of experience using a white cane for mobility. Personalized adjustments in walking planning and feedback intensity were made based on individual habits and preferences.

TABLE II
THE STATISTICAL RESULTS OF A GUIDANCE EXPERIMENT INVOLVING 10 LIMITED VISION INDIVIDUALS($^{*}p < 0.05$, $^{**}p < 5 \times 10^{-3}$, *and* $^{***}p < 5 \times 10^{-4}$)

	Duration(s)			Contact Times		
	2P Touch	A*	WPG	2P Touch	A*	WPG
Avg	23.56	16.42	**10.28**	6	2.57	**0.77**
Std	7.36	1.98	1.48	1.06	0.9	0.9
Max	31.62	21.4	15.2	8	4	3
2P Touch	-	*	***	-	*	***
A*	*	-	**	*	-	**
WPG	***	**	-	***	**	-

Three random obstacles were placed in a narrow corridor measuring 10 meters long and 2 meters wide. The obstacles included low-height and hanging obstacles (refer to Figure 11).

In the experiment, ten participants were assigned Two-Point Touch, the A* algorithm guidance, and our proposed WPG guidance method in a randomized order. Two point touch refers to a method used by limited vision individuals while navigating with a white cane, involving the cane touching the ground in two places alternately to detect obstacles and gather information about the surrounding environment. All participants were instructed to start from the same point and navigated the 10-meter narrow corridor to reach the finishing line. Due to safety considerations, the positions of the obstacles were not changed.

Using the faster-LIO feature of Smart Cane V2, we accurately recorded trajectories. The collected data, as presented in Table II, revealed that the WPG navigation method displayed the best performance with the lowest average duration and collision frequency. These findings were consistent with results obtained from experiments involving blindfolded novice participants.

The Two-Point Touch method, while designed for safety, resulted in longer task completion times and a higher frequency of collisions, leading to task failure in some instances. Comparatively, both the A* and WPG methods incorporated collision detection mechanisms that effectively helped participants avoid obstacles. The A* and WPG methods showed a minor difference in average duration. However, a substantial contrast was observed in collision incidents, with WPG at 0.7 incidents and A* at 2.1 incidents. WPG demonstrated a stronger capability to prevent collisions, thus providing an obvious advantage over A*.

The statistical results of the guidance experiment are shown in Table II. The average duration of the WPG method was 10.28 seconds, while the A* method was 16.42 seconds. The results of the ANOVA model ($^{*}p < 0.05$,$^{**}p < 5 \times 10^{-3}$, *and* $^{***}p < 5 \times 10^{-4}$) indicated that the effect of using the WPG method or the A* method on the participants was significant. Furthermore, the Tukey HSD test (alpha=0.05) showed that the WPG method was significantly different from the A* method.

Further analysis indicated that A* navigation induced larger heading angle variations, while WPG, considering human kinematic constraints, managed to minimize significant alterations in heading angles. This would contribute to the superior collision avoidance capability of WPG compared to that of A*.

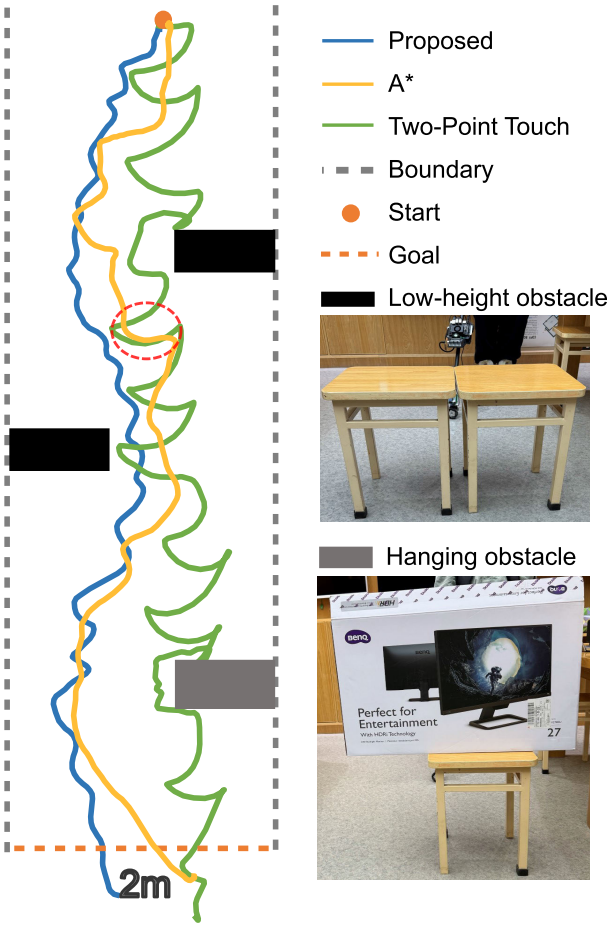

Fig. 11. This is a trajectory comparison chart of a participant completing the guidance experiments evaluation. For WPG and A* guidance, the trajectory of the 'Human-cane system' state. For the 2P Touch method, the trajectory represents the cane's center point trajectory. The red marks indicate areas where A* generates significant changes in heading angles.

VI. CONCLUSION

This paper presents Smart Cane, a comprehensive solution designed to assist visually impaired individuals in navigation. The developed software and hardware framework provides a robust platform for addressing mobility challenges faced by the visually impaired. The main contribution is the development of a method for generating walking paths. The proposed approach combines the utilization of a LIPM and LFPC motion primitives to generate walking paths specifically designed for visually impaired individuals. By comparing our proposed method with the A* evaluation index in a simulation environment, we demonstrated its superiority in generating smaller-scale paths, leading to smoother navigation for visually impaired individuals. In addition, our findings highlight the advantages of the Smart Cane over traditional white canes, including reduced duration, fewer contacts/collisions, and improved path length consistency. It minimized heading angle variations compared to A*, resulting in smoother trajectories for navigation.

Our experimental design simulated certain real-world navigation scenarios in a controlled setting. However, the guidance experiments lacked consideration for dynamic objects, and the positions of obstacles remained unchanged throughout the trials. This simplification of real-world conditions is a limitation of our study, and future research should aim to include these factors for a more comprehensive evaluation.

Our forthcoming research endeavors seek to augment the Smart Cane's functionalities, specifically targeting real-time environmental perception within complex, dynamic scenarios encompassing diverse elements. Our future work aims to design algorithms of walking path generation in 3D environment by addressing diverse ground irregularities. Moreover, we also acknowledge that further research is needed to address the challenges of dynamic shifting obstacles.

VII. Acknowledgment

The authors declare no competing financial interests. The authors extend their gratitude to the study participants for their invaluable contributions. Additionally, they would like to thank the reviewers for their valuable insights and constructive feedback.

References

[1] J. D. Steinmetz et al., "Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: The right to sight: An analysis for the global burden of disease study," *Lancet Global Health*, vol. 9, no. 2, pp. e144–e160, 2021.

[2] T. Xu et al., "Prevalence and causes of vision loss in China from 1990 to 2019: Findings from the global burden of disease study 2019," *Lancet Public Health*, vol. 5, no. 12, pp. e682–e691, Dec. 2020.

[3] C. Ye, S. Hong, X. Qian, and W. Wu, "Co-robotic cane: A new robotic navigation aid for the visually impaired," *IEEE Syst. Man, Cybern. Syst. Mag.*, vol. 2, no. 2, pp. 33–42, Apr. 2016.

[4] R. K. Katzschmann, B. Araki, and D. Rus, "Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 583–593, Mar. 2018.

[5] C. Ye and X. Qian, "3-D object recognition of a robotic navigation aid for the visually impaired," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 441–450, Feb. 2018.

[6] W. Balachandran, F. Cecelja, and P. Ptasinski, "A GPS based navigation aid for the blind," in *Proc. 17th Int. Conf. Appl. Electromagn. Commun. (ICECom)*, Oct. 2003, pp. 34–36.

[7] N. Shalal et al., "A review of autonomous navigation systems in agricultural environments," in *SEAg 2013: Innovative Agricultural Technologies for a Sustainable Future*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:35147407

[8] J. Aulinas et al., "The slam problem: A survey," in *Artificial Intelligence Research and Development*, 2008, pp. 363–371, doi: 10.3233/978-1-58603-925-7-363.

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[10] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[11] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory Comput.*, vol. 8, no. 1, pp. 415–428, 2012.

[13] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath, "Robotic guide dog: Leading a human with leash-guided hybrid physical interaction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11470–11476.

[14] P. Slade, A. Tambe, and M. J. Kochenderfer, "Multimodal sensing and intuitive steering assistance improve navigation and mobility for people with impaired vision," *Sci. Robot.*, vol. 6, no. 59, Oct. 2021, Art. no. eabg6594.

[15] S. Agrawal, M. E. West, and B. Hayes, "A novel perceptive robotic cane with haptic navigation for enabling vision-independent participation in the social dynamics of seat choice," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 9156–9163.

[16] A. A. Masoud, "Kinodynamic motion planning," *IEEE Robot. Autom. Mag.*, vol. 17, no. 1, pp. 85–99, Mar. 2010, doi: 10.1109/MRA.2010.935794.

[17] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[18] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, Apr. 2010.

[19] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.

[20] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation," *J. Field Robot.*, vol. 37, no. 8, pp. 1300–1313, Dec. 2020.

[21] A. V. Papadopoulos, L. Bascetta, and G. Ferretti, "Generation of human walking paths," *Auto. Robots*, vol. 40, no. 1, pp. 59–75, Jan. 2016.

[22] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—An inverse optimal control approach," *Auto. Robots*, vol. 28, no. 3, pp. 369–383, Apr. 2010.

[23] D. Clever and K. D. Mombaur, "An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots," in *Robotics: Science and Systems*, vol. 12, 2016, doi: 10.15607/rss.2016.xii.005.

[24] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*, vol. 101. Cham, Switzerland: Springer, 2014.

[25] L. Ye, X. Wang, H. Liu, and B. Liang, "The simplest balance controller for dynamic walking," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2022, pp. 1793–1800.

[26] S. Kajita et al., "Biped gait control based on spatially quantized dynamics," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2018, pp. 75–81.

[27] K. Yin, K. Loken, and M. Van de Panne, "Simbicon: Simple biped locomotion control," *ACM Trans. Graph. (TOG)*, vol. 26, no. 3, pp. 10–105, 2007.

[28] J. Ackermann et al., *Robust Control: Parameter Space Approach*, vol. 2. Cham, Switzerland: Springer, 2002.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SCS-4, no. 2, pp. 100–107, Jul. 1968.

[30] C. Cao et al., "Autonomous exploration development environment and the planning algorithms," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8921–8928.