Real-Time Optimisation-Based Path Planning for Visually Impaired People in Dynamic Environments

Hadeel R. Surougi, Julie A. McCann
Imperial College London
London, UK, SW7 2AZ
{h.surougi19, j.mccann}@imperial.ac.uk

Abstract

Most existing outdoor assistive mobility solutions notify Visually Impaired People (VIP) about potential collisions but fail to provide Optimal Local Collision-Free Path Planning (OLCFPP) to enable the VIP to get out of the way effectively. In this paper, we propose MinD, the first VIP OLCFPP scheme that notifies the VIP of the shortest path required to avoid Critical Moving Objects (CMOs), like cars, motorcycles, etc. This simultaneously accounts for the VIP's mobility constraints, the different CMO types and movement patterns, and predicted collision times, conducting a safety prediction trajectory analysis of the optimal path for the VIP to move in. We implement a real-world prototype to conduct extensive outdoor experiments that record the aforementioned parameters, and this populates our simulations for evaluation against the state-of-the-art. Experimental results demonstrate that MinD outperforms the Artificial Potential Field (APF) approach in effectively planning a short collision-free route, requiring only $1.69m$ *of movement on average, shorter than APF by* 90.23%*, with a* 0% *collision rate; adapting to the VIP's mobility limitations and provides a high safe time separation (*$> 5.35s$ *on average compared to APF). MinD also shows near real-time performance, with decisions taking only* $0.04s$ *processing time on a standard off-the-shelf laptop.*

1. Introduction

Visually Impaired People (VIP) face challenges in navigating outdoor environments independently, as their limited perception hinders their awareness of autonomous moving objects, including vehicles, bicycles, and pedestrians, which pose different risks to VIPs' physical safety [29]. According to [21], more than 30,000 VIPs in India die annually and around 400 VIPs are injured due to pedestrian-vehicle accidents. Demonstrated to be of great significance, existing VIP mobility assistive methods can be classified into traditional tools (*e.g.,* white canes and guide dogs) and non-vision/vision-based solutions [10]. Research has favoured vision-based solutions due to the potential to provide richer spatial and contextual information. Most vision-based VIP mobility assistive solutions successfully detect obstacles [5, 9, 19]. However, they significantly fail to assist VIPs in effectively avoiding collisions as they do not carry out *optimal collision-free path planning*.

Path planning finds a route from a current location to a final destination, while avoiding obstacles in between [1]. It is crucial for efficient VIP navigation, as it evaluates all navigation possibilities to avoid unsafe or unnecessary routing while minimizing VIP's movement time, effort, and computer resources [22]. Consequently, many efforts have been made to improve VIP mobility through either borrowing existing robot global path planning algorithms (*e.g.,* A* [3], Dijkstra [33], Ant Colony Optimisation [22], and Particle Swarm Optimisation [34]) or proposing novel ones [23, 20]. However, such global path planning algorithms cannot prevent VIP collisions with moving objects as they require static representation of an environment, which is impractical for real-world scenarios.

Compared to global path planning, local path planning is more promising as it can efficiently respond to real-time changes in VIP's surroundings, such as obstacles suddenly appearing, by dynamically calculating local collision-free routes without requiring a full predefined environmental map. Existing local path planning algorithms are mainly used for robotics [4]; however, they cannot be directly applied to VIPs as robot mobility constraints are different. Unlike VIPs, robots are fully controllable; their avoidance reaction (*e.g.,* stop/move) and motion (*e.g.,* acceleration/velocity) are automatically controlled and adjusted by planning algorithms [24], which makes such algorithms more complex. Further, ensuring VIP safety is with a top priority, and mistakes cannot be tolerated. This requires a shift in perspective that considers VIPs' physical abilities in practice. As far as we know, no study has developed a local path planning algorithm dedicated to VIPs that can

be applied to an unknown outdoor environment containing moving objects of various classes and speeds.

Therefore, this paper presents a novel local path planning scheme, *MinD*, to assist VIPs in safely navigating outdoor environments in which different classes (*i.e.,* car, motorcycle, bicycle, and pedestrian) of objects moving at varying speeds are present. *MinD* aims to minimise the distance a VIP needs to take to dodge such *Critical Moving Objects* (CMOs), while avoiding unnecessary transition distances. To the best of our knowledge, *MinD* is the *first* VIP-specific local path planning scheme that addresses real-time VIP navigation assistance considering VIPs limited physical abilities (*i.e.,* walking speed and auditory perception), ensuring enough time for VIPs to avoid CMOs effectively, and is able to provide look-ahead safety prediction. This work is part of our larger project to develop a VIP-navigation assistive system for dynamic environments. Contributions of this paper can be summarized as follows.

Contribution 1. We develop a collision-free path planning scheme consistent with VIPs' limited capabilities, which plans the shortest path required by the VIP to avoid CMOs, where the VIP's walking speed, auditory perception, various CMO types, and CMOs' movement patterns are comprehensively accounted for to ensure that the VIP can avoid threats effectively and within a safe time. It also conducts a look-ahead safety prediction trajectory analysis of the optimal path to ensure more safety for VIPs. This predicts the VIP's and CMO's future positions and movement patterns, and estimates the collision-free and safe time separation between the VIP and CMO along the optimal path according to an adaptive threshold set based on their speeds.

Contribution 2. The VIP-local path planning problem is challenged by being non-convex by nature. Therefore, we re-formulate the non-convex problem into a solvable convex local path planning sub-problem using *Sequential Convex Programming* (SCP) to find a tolerable sub-optimal solution and reduce the computational complexity of the original non-convex problem.

Contribution 3. To evaluate our scheme in a highly practical setting, we implement a real-world prototype, using commercially available devices, *i.e.,* an RGB camera, a laptop, and headphone, and carry out extensive experiments with people walking, cycling and driving in outdoor spaces, while a researcher mimics the VIP - from this, parameters were derived, which were populate our simulator parameters. Note, simulation allows safe 'what-if' experimentation to fully evaluate our algorithm against the state-of-the-art, and in an ethically safe, near-to-realistic way. Experimental results show that *MinD* outperforms the closest state-of-the-art *Artificial Potential Field* (APF)[16] approach and the baseline Notification-Based Random Decision (NBRD) method in effectively planning free-collision paths suitable for VIP's mobility and perception conditions ensuring short avoidance distances. *MinD* requires only $1.69m$ of avoidance distance on average for all CMO classes moving at varying speeds from $3.6km/h$ to $108km/h$, which is considerably shorter than APF and NBRD by 90.23% and 81.04%, respectively, and ensures a 0% collision rate. With this distance, the VIP can avoid a CMO moving up to $108km/h$ with only $1.22s \sim 4.05s$ reaction time when walking at speeds of $1.5km/h \sim 5km/h$. It also achieves a large average safe time separation ($11.52s$ for cars, $11.59s$ for motorcycles, $8.69s$ for bicycles and $6.68s$ for pedestrians, which is $> 5.35s$ and $> 1.23s$ compared to APF and NBRD, respectively), with only $0.04s$ processing speed.

2. Related Work

This section discusses previous work on local path planning in robotics and autonomous vehicles.

An Unmanned Aerial Vehicles (UAVs) collision avoidance approach is proposed in [30], in which Simulated Annealing (SA) and a simple threat-avoidance method are combined to find a near-optimal path in a 2D radar environment constrained by regular circular threats. While the algorithm successfully escapes form local minima and produces threat-free paths, it suffers from slow convergence, resulting in long computation times, *i.e.,* it requires $921,030$ iterations and $9.218s$ to find the best solution in a 2D map with 10 threats, using an Intel Core2 Duo P8600 $2.4GHz$ CPU. Further, to produce a threat-free path, it considers only the circular areas occupied by the threats, ignoring their motion (*i.e.,* moving speed and direction).

In [12], a Velocity Obstacle (VO) algorithm is proposed for robot motion planning in dynamic environments. VO defines a set of a robot velocities that lead to a collision with an object moving at a given speed. Then, the set is filtered out to produce all possible velocities that avoid collision with the obstacle, which is reduced to the dynamically feasible manoeuvres based on the robot's acceleration constraint. VO is effective in avoiding dynamic obstacles, and several improvements have been proposed, *e.g.,* Reciprocal Velocity Obstacle (RVO) [31] and ClearPath [13]. Although VO-based methods have demonstrated their effectiveness in assisting robots to avoid a collision, it is difficult to apply them to VIPs as we cannot control their speeds as definitively. Similarly, in [26] and [18], an algorithm-based Artificial Potential Field (APF) is proposed. APF is inspired by a physical concept, a potential field, where the robot moves as a result of two forces: the destination's gravity force and the obstacle's repulse force. APF is simple and able to work in real-time. Nevertheless, it is prone to failure in cases where the repulse and gravity efforts are equal; thus, the autonomous agent never reaches the destination.

A human-centred X–Y–T space path planning is proposed in [25] to assist robots mobile safely in dynamic environments and to provide free space to pedestrians in the

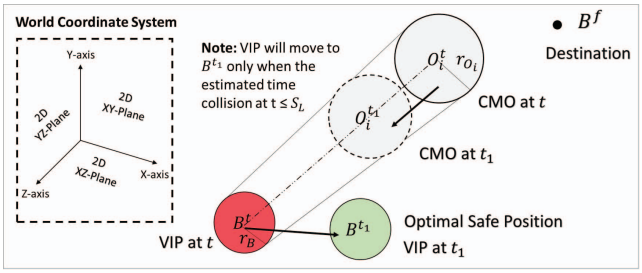


Figure 1. VIP Local Path Planning Problem.

same vicinity. Based on a 2D grid-based representation of the surroundings, it dynamically produces collision-free path motion considering the robot's dynamic constraints and personal human space and directional area. However, storing environmental information in a grid-based map requires significant storage space, which can negatively impact the algorithm's efficiency and scalability when dealing with larger environments. In [11], a Deep Reinforcement Learning-based path planning algorithm is proposed to assist robots in navigating safely among pedestrians. It is an extended version of [7] and [6], which learns collision avoidance without assuming that pedestrians follow any particular behaviour rule. However, for training, DRL requires a large-scale dataset, which may not always be available and needs to generate training data from extensive random simulation scenarios, and consumes high resources and time. Besides, it cannot guarantee an effective performance on real-world data that differs from the training data.

3. Path Planning Problem Description, Formulation and Approximation

3.1. Problem Description

Since we aim to assist VIPs in avoiding CMOs of different classes, *i.e.,* cars, motorcycles, bicycles, and pedestrians, in unfamiliar dynamic environments, our problem is classified as *local path planning*. To address this problem effectively, we need to detect CMOs and estimate their current and future movement, *i.e.,* position and velocity, and VIP's corresponding trajectories.

As shown in Fig. 1, we assume that the VIP occupies a circular zone C_B with a centre at point B^t and a radius of r_B at time t on the World Coordinate System [1](WCS). In our problem, the VIP moves forward towards a known final distention B^f at a velocity of $\boldsymbol{v_B} = |\boldsymbol{v_B}|\angle\beta$, where $|\boldsymbol{v_B}|$ is the speed, and $\angle\beta$ is the moving angle[2]. We also assume that a CMO is at point O_i^t, the center of the circle C_{O_i} with a radius of r_{O_i}, and moves to $O_i^{t_1}$ at velocity of $\boldsymbol{v_{O_i}} = |\boldsymbol{v_{O_i}}|\angle\theta$. Then, The VIP will be guided to move to B^{t_1} at t_1 only when a collision is detected (the CMO heads toward the VIP, determined by our algorithm in Appendix A) and the estimated collision time between the CMO and the VIP at t exceeds a certain predetermined threshold S_L. When multiple CMOs exist, we prioritize the nearest threat to be avoided by considering the CMO with the least collision time, see Eq. (1). To ensure the VIP's physical safety, the r_B and r_{O_i} should cover the full width of the VIP and CMO, respectively.

[1]It is a real-world 3D Cartesian coordinate system with a predefined origin.

[2]The angle between the VIP moving direction and the Z-axis of the WCS.

3.2. Problem Formulation and Approximation

VIPs have limited physical abilities, such as walking at a limited speed (*i.e.,* $\leq 5km/h$[8]) and relying on their sense of hearing to perceive their surroundings. Therefore, planning a safe path in advance and providing enough time for VIPs to receive acoustic guidance instructions and react safely, *i.e.,* follow the proposed path, is essential for their safety, particularly in the presence of high-risk CMOs like cars and motorcycles. Avoiding unnecessary transitions of VIPs with low-risk CMOs such as bicycles and pedestrians enhances algorithm usability. Accordingly, we estimate the minimum VIP-CMO collision time to activate our path planning algorithm as:

$$min(\frac{||O_i^t - B^t||_2}{|v_{O_i}^t| + |v_B^t|}) \leq S_L, i = [1, .., n], \qquad (1)$$

where n is the number of the CMOs, S_L is a predefined time threshold (in s) where its value is adapted based on the class and speed of CMO i. The $B^t = (X, Y, Z)_B^t$ and $O_i^t = (X, Y, Z)_{O_i}^t$ are the *current* 3D positions of the VIP and CMO i at time t, respectively, in the WCS. The $v_B^t = (v_X, v_Y, v_Z)_B^t$ and $v_{O_i}^t = (v_X, v_Y, v_Z)_{O_i}^t$ are the *current* velocities of the VIP and CMO i at t, where v_X, v_Y, and v_Z are the velocity components of the X-axis, Y-axis, and Z-axis of the WCS, respectively, and they are defined as: $v_B^t = (\frac{X_B^t - X_B^{t_0}}{t-t_0}, \frac{Y_B^t - Y_B^{t_0}}{t-t_0}, \frac{Z_B^t - Z_B^{t_0}}{t-t_0})_B^t$ and $v_{O_i}^t = (\frac{X_{O_i}^t - X_{O_i}^{t_0}}{t-t_0}, \frac{Y_{O_i}^t - Y_{O_i}^{t_0}}{t-t_0}, \frac{Z_{O_i}^t - Z_{O_i}^{t_0}}{t-t_0})_{O_i}^t$, where $(X, Y, Z)_B^{t_0}$ and $(X, Y, Z)_{O_i}^{t_0}$ are the 3D coordinates of the *previous* positions (B^{t_0} and $O_i^{t_0}$) of the VIP and CMO i at t_0, respectively. We use the sum of the absolute values of the VIP and CMO velocities, assuming a worst-case analysis whereby the VIP and CMO are directly heading to each other. This is to provide VIPs with a large safety margin (as possible).

When the least estimated collision time exceeds S_L as in Eq.(1), our objective is to plan the shortest path that ensures the safety of a VIP by avoiding CMOs of various classes and varying speeds, considering the aforementioned VIPs' limited capabilities. To achieve this, we optimise the next position B^{t_1} for the VIP, subject to six practical constraints that take into account the VIP's abilities and the dynamic states of CMOs as follows.

C_1: The *maximum travelling distance* that the VIP can travel while avoiding the CMO. This constraint is set to consider the VIPs' limited mobility and prioritize their proximity to the initial path to avoid getting lost or straying from their destination. Mathematically, it can be expressed as:

$$||B^{t_1} - B^t||_2 \leq D_{max}, \tag{2}$$

where $B^{t_1} = (X, Y, Z)^{t_1}_B$ is the *optimal safe* 3D position of the VIP at t_1. The D_{max} is a predefined maximum distance threshold in m.

C_2: The *maximum travelling time* that is required by the VIP to safely reach B^{t_1} based on their moving speed - note this parameter is variable and can be tailored to the user from mobile phone walking measurements. Mathematically, this constraint can be expressed as:

$$||B^{t_1} - B^t||_2 \leq T_{max}|v^{t_1}_B|. \tag{3}$$

where the $v^{t_1}_B$ is the *next* velocity of the VIP at t_1, and the T_{max} is the maximum expected time (in s) the VIP requires to successfully travel from B^t to B^{t_1}.

C_3: More than merely moving the VIPs to an obstacle-free position is required to ensure their safety, as CMOs can move at varying speeds and suddenly change their direction. Therefore, it is essential to predict the CMO's *next* position $O^{t_1}_i$ and estimate the time separation between the VIP and CMO positions at t_1. Besides, this time separation should be sufficiently large to guarantee more safety for the VIP, especially, when the CMO shifts towards the VIP, who begins evading. Based on this, we formulate the *safe time separation* constraint as follows:

$$||O^{t_1}_i - B^{t_1}||_2 > s_l(|v^{t_1}_{O_i}| + |v^{t_1}_B|), \tag{4}$$

$$O^{t_1}_i = O^t_i + v^t_{O_i}(t_1 - t), \tag{5}$$

where $v^{t_1}_{O_i}$ is the $next$ velocity of the CMO i at t_1, and s_l is the predefined minimum time interval (in s) that must be maintained between B^{t_1} and $O^{t_1}_i$. The s_l value is adjusted based on the CMO i's class and speed, such that higher-risk CMOs (*i.e.,* cars and motorcycles) require a larger s_l to ensure adequate safety for the VIP.

To successfully avoid collision with the CMO, the VIP must move outside the *critical zone*, a potential threat area (shadowed in Fig. 2) confined between the two tangents $\overleftrightarrow{O_1B_1}$ and $\overleftrightarrow{O_2B_2}$ of circles C_B and C_{O_i} intersected at point A. To achieve this, we define two *collision avoidance* constraints (C_4 and C_5) as follows. Note, we only consider the 2D XZ-plane of the WCS (Fig. 1), the ground plane shared by the VIP and the CMO as we assume that camera movement along the Y-axis is negligible as it is perpendicular to the ground plane.

C_4: To change the VIP's direction away from the CMO's path, the *moving angle* constraint is imposed, which is expressed as follows:

$$\beta < (\phi - \alpha) - \delta_a, \tag{6}$$

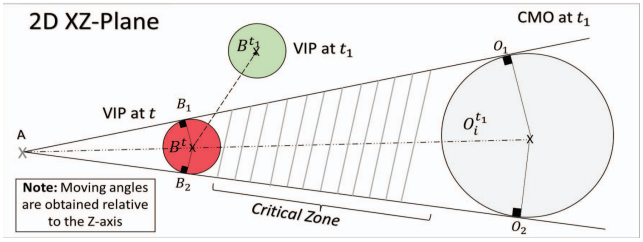


Figure 2. CMOs Collision Avoidance.

$$\alpha = tan^{-1}\left(\frac{1}{\sqrt{\frac{d^2}{(r_{O_i} - r_B)^2} - 1}}\right), \tag{7}$$

$$\phi = tan^{-1}\left(\frac{X^{t_1}_{O_i} - X^t_B}{Z^{t_1}_{O_i} - Z^t_B}\right), \tag{8}$$

$$\beta = tan^{-1}\left(\frac{X^{t_1}_B - X^t_B}{Z^{t_1}_B - Z^t_B}\right), \tag{9}$$

where α is the angle between $\overline{AO^{t_1}_i}$ and $\overline{AO_1}$ (see Fig. 2 and for the derivations, see Appendix B), $d = |\overline{B^t O^{t_1}_i}|$ is the Euclidean distance between B^t and $O^{t_1}_i$ and can be calculated as $d = \sqrt{(X^{t_1}_{O_i} - X^t_B)^2 + (Z^{t_1}_{O_i} - Z^t_B)^2}$, ϕ is the angle of the displacement vector from B^t to $O^{t_1}_i$ (relative to the Z-axis of the WCS), and β is the angle of the displacement vector from B^t to B^{t_1}. The δ_a is an *angle safety margin* to ensure that the VIP moves away sufficiently from the CMO's direction of motion, especially when CMOs are detected early at a far distance since the object may appear smaller at greater distances, resulting in a smaller alpha value. The δ_a can be adapted based on the VIP safety requirements. The equivalent expression of Eq. (6) is:

$$(X^{t_1}_B - X^t_B) - (tan((\phi - \alpha) - \delta_a)(Z^{t_1}_B - Z^t_B)) < 0. \tag{10}$$

Another form of the *moving angle* constraint is:

$$(X^{t_1}_B - X^t_B) - (tan((\phi + \alpha) + \delta_a)(Z^{t_1}_B - Z^t_B)) > 0. \tag{11}$$

Eq. (10) and Eq. (11) can be used interchangeably to move the VIP to the left or right based on the CMO relative position, respectively.

C_5: To ensure that the VIP moves outside of the threat area (the *critical zone* in Fig. 2), a constraint on the VIP's *moving distance* can be applied as:

$$||B^{t_1} - B^t||_2 > (r_B + r_{O_i}) + \delta_d, \tag{12}$$

where δ_d is a *distance safety margin* (can be adapted according to VIP safety requirements) that ensures that the VIP moves outside the critical zone with sufficient distance. This is because the optimal paths may be diagonal lines rather than horizontal ones. Besides, objects do

not always move in straight lines; their paths may involve slight changes in their moving direction. This may lead the scheme to detect multiple collisions between the VIP and the same object within a short period of time. Therefore, δ_d can prevent the VIP from being transited frequently, that is, actually unnecessary.

C_6: To ensure that VIP continually advances toward the final destination B^f, we define the *forward moving* constraint as follows:

$$||B^{t_1} - B^f||_2 \leq ||B^t - B^f||_2. \tag{13}$$

Based on all above definitions, we formulate a non-convex VIP-specific local path planning problem, the *non-convex MinD*, as follows:

$$\min_{B^{t_1}} f(B^{t_1}) = ||B^{t_1} - B^t||_2| \left\{ \min(\frac{||O_i^t - B^t||_2}{|v_{O_i}^t| + |v_B^t|}) \leq S_L \right\}$$

s. t.

C_1: Eq. (2), C_2: Eq. (3), C_3: Eq. (4), C_4: Eq. (10),
C_5: Eq. (12), and C_6: Eq. (13) (14)

The objective function $f(B^{t_1})$ and all the constraints in Eq. (14) are convex except C_3 and C_5; both are non-convex constraints due to the violation of the standard formulation of the convex optimisation problem (*i.e.,* $C(x) \leq 0$[2]). For this reason, the *MinD* becomes a non-convex problem. In the absence of convexity, a problem cannot be solved.

To solve the *non-convex MinD* problem in Eq. (14), we use the *Sequential Convex Programming* (SCP) [2]. In SCP, the original non-convex problem is transformed into a series of convex sub-problems, by replacing the non-convex constraint or objective function with convex approximations around an approximation point, that can be iteratively solved, using a convex algorithm, starting from an initial guess until it converges to a solution that is acceptable with certain convergence criteria, see Algorithm 1. Therefore, we approximate the non-convex constraints C_3 and C_5 to convex constraints using the *first-order Taylor* series approximation. In the following, the equivalent convex constraints C_3' and C_5' of the C_3 and C_5, respectively:

$$C_3' = ||O_i^{t_1} - B_k^{t_1}||_2 + [\frac{-(X_{O_i}^{t_1} - X_{B_k}^{t_1})}{||O_i^{t_1} - B_k^{t_1}||_2}, \frac{-(Y_{O_i}^{t_1} - Y_{B_k}^{t_1})}{||O_i^{t_1} - B_k^{t_1}||_2}, \frac{-(Z_{O_i}^{t_1} - Z_{B_k}^{t_1})}{||O_i^{t_1} - B_k^{t_1}||_2}](B^{t_1} - B_k^{t_1}) > s_l(|v_{O_i}^{t_1}| + |v_B^{t_1}|), \tag{15}$$

$$C_5' = ||B_k^{t_1} - B^t||_2 + [\frac{(X_{B_k}^{t_1} - X_B^t)}{||B_k^{t_1} - B^t||_2}, \frac{(Y_{B_k}^{t_1} - Y_B^t)}{||B_k^{t_1} - B^t||_2}, \frac{(Z_{B_k}^{t_1} - Z_B^t)}{||B_k^{t_1} - B^t||_2}](B^{t_1} - B_k^{t_1}) > (r_B + r_{O_i}) + \delta_d, \tag{16}$$

where $B_k^{t_1}$ is an approximation point that is used to approximate a solution around, and it can be initially set to any random value that produces a feasible solution to the approximated sub-problem. Based on the above approximations, we re-formulate the Eq. (14) to the following:

$$\min_{B^{t_1}} f(B^{t_1}) = ||B^{t_1} - B^t||_2| \left\{ \min(\frac{||O_i^t - B^t||_2}{|v_{O_i}^t| + |v_B^t|}) \leq S_L \right\}$$

s. t.

C_1: Eq. (2), C_2: Eq. (3), C_3': Eq. (15), C_4: Eq. (10),
C_5': Eq. (16), and C_6: Eq. (13) (17)

To solve the convex sub-problem in Eq. (17), we set a random initial value $B_0^{t_1}$, and define a tolerance variable $\epsilon > 0$ as a convergence criteria to stop the iteration k and find the global optimal solution B^{t_1}. The algorithms of solving Eq. (17) is summarized in Algorithm 1.

Algorithm 1 The SCP of the convex *MinD* sub-problem

```
Given initial point B_0^{t1}, iteration k, and tolerance ε > 0
for i = 0, 1, ..., k do
    Solve the convex sub-problem in Eq. (17)
    if Eq. (17) is feasible then
        B_{k+1}^{t1} ← B^{t1}
        if ||B_{k+1}^{t1} − B_k^{t1}||_2 < ε then
            Stop it is converged
        else
            B_k^{t1} ← B_{k+1}^{t1}
            Continue
        end if
    else
        Initialize a new point B_k^{t1}
    end if
end for
```

4. Evaluation

To evaluate our work in a highly practical setting, we implemented a real-world prototype, see Fig. C.2 in Appendix C, using commercially available devices (*i.e.,* a camera, laptop, and headphones). The specifications of such devices and implementation details are presented in Appendix C.

With our prototype, we conducted measurements in structured wide outdoor environments and captured videos of lightly choreographed (follow predetermined scenarios) volunteers driving cars, motorcycles and bicycles and walking at different speeds while a person mimics a VIP. Then, we analyze these objects' movements, calculate collision times, and studied the emission deadlines of early alarms for the VIP to avoid CMOs effectively (details are provided in Appendix C). This is to accurately parameterize our simulator and set practical parameters (*e.g.,* r_B and T_{max} for VIP,

and r_O, δ_d, S_L, s_l, for each CMO class) that would map a real-world implementation (*i.e.,* by leaving enough time for the VIP to react, especially with high-speed objects). Simulation allows safe 'what-if' experimentation to fully evaluate *MinD* effectively and in a near-to-realistic way.

4.1. Empirically Parameterized Simulator Setup

To this end, we developed a 2D Python-based simulator to randomly create large numbers of experimental scenarios and simulate CMO's movements and the VIP's manoeuvres using basic geometric algorithms without compromising health and safety, where its parameters were learned from our real-world experiments. The simulator used Ubuntu 14.04.4 LTS, with an Intel Core i9-9900K CPU and 64GB RAM. We use a 2D simulation environment because we assume the Y-axis coordinates of the VIP and CMO are zeros since the Y-axis is vertical to the ground plane (2D XZ-plane, the shared plane between the VIP and CMO, see Fig. 1). To effectively evaluate our proposed algorithm, we used the Monte Carlo method [27] in our simulation experiments. In particular, for each class, we moved the CMO towards the VIP at 8 different speeds, $11km/h \sim 108km/h$ for cars and motorcycles, $7km/h \sim 54km/h$ for bicycles, and $3.4km/h \sim 36km/h$ for pedestrians. Then, for each speed, we ran 100 samples with random CMO's positions. This means 3200 samples for all CMO classes (800 for each class) are included in the experiments. The environments we simulate are car parks, near driveways, pedestrian crossings and walking footpaths, with a dimension of $20m$x$500m$, with the initial VIP's starting position at $B_t^* = (0, 0)$ and final destination at $B^f = (0, 500)$. For simplicity, we assume that the CMO, in all our experiments, moves directly toward the VIP without changing its direction and speed (*i.e.,* $|V_O^{t_1}| == |V_O^t|$). We expect that the VIP can change his/her direction to avoid such threats; however CMOs adapting their speed and direction are left for further work. We also assume that the VIP walks at a fixed speed of 5km/h ($|V_B^{t_1}| == |V_B^t|$), and moves forward towards the predefined known final destination B^f. Further work will look at the VIP adapting speed as well.

For parameter setting, we differentiate between CMOs classes in terms of risk level (*i.e.,* cars and motorcycles are high-risk CMOs, and bicycles and pedestrians are low-risk CMOs). The parameter settings that are adjusted for each CMO's class, are as follows, car: $S_L = 12s$, $s_l = 5s$, $r_O = 0.9m$, and $\delta_d = 1.1m$, motorcycle: $S_L = 12s$, $s_l = 5s$, $r_O = 0.45m$, and $\delta_d = 0.65m$, bicycle: $S_L = 9s$, $s_l = 4s$, $r_O = 0.35m$, and $\delta_d = 0.55m$, and pedestrian: $S_L = 7s$, $s_l = 3s$, $r_O = 0.27m$, and $\delta_d = 0.47m$. For cars and motorcycles, we set larger threshold (*i.e.,* S_L and s_l) values than bicycles and pedestrians to guarantee more safety for the VIP by early planning a safe path to follow, considering a VIP's limited capabilities (auditory perception and slow moving speed), and leaving a greater time separation between the VIP and CMO at t_1. Since bicycles and pedestrians are less dangerous, we reduced the values to avoid unnecessary transitions for the VIP. For all CMO classes, we set the other parameters as: $B_t^* = (0m, 0m)$, $B_f = (0m, 500m)$, $r_B = 0.25m$, $D_{max} = 10m$, $T_{max} = 5s$, $\delta_a = 30°$, $k = 100$, and $\epsilon = 0.001$.

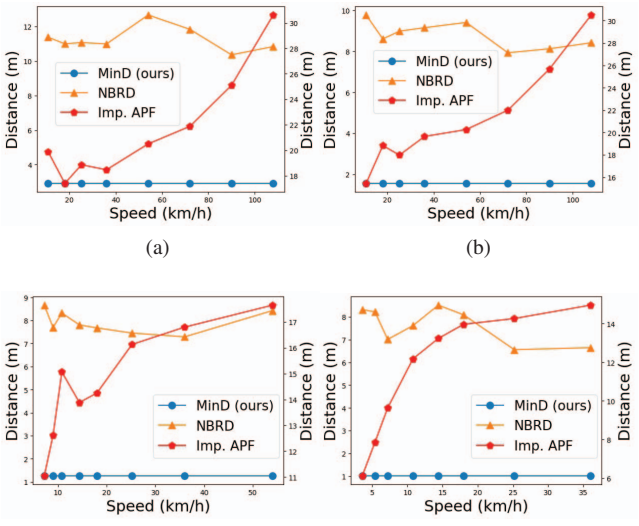
4.2. Evaluation Metrics

For evaluation, we define **four** metrics as follows. First, *travelling distance* to measure the path length that the VIP should travel to avoid threats, calculated as: $||B^{t_1} - B^t||_2$. Second, *safe time separation* between the VIP and CMO in the optimal path to measure its safety, calculated as: $\frac{||O_i^{t_1} - B^{t_1}||_2}{|v_{O_i}^{t_1}| + |v_B^{t_1}|}$. Third, *collision rate* to measure our scheme performance in planning a collision-free path. In collision rate, we consider a collision if the *safe time separation* is less than sl that is set based on CMO risk level, see Section 4.1. Then we calculate the collision rate as: $\frac{1}{m}\sum_{i=1}^{m}(\frac{C}{N})$, where m is the number of the samples that represent CMO moving speeds for a particular CMO class (*i.e.,* 8 speeds), N is the number of random samples running under each CMO moving speed (*i.e.,* 100 samples), and C is the number of collisions. Fourth, *computation time* and *convergence iteration* to measure the efficiency of *MinD* in providing real-time path planning.

4.3. Evaluation Methodology

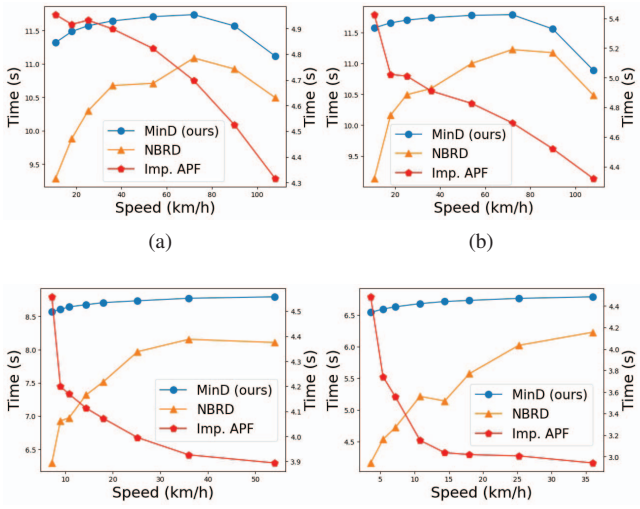
To investigate our algorithm's performance in the absence of specific-VIP local path planning algorithms, we compared our algorithm with two methods using the above metrics. The first method is a baseline, a *Notification-Based Random Decision* (NBRD) method. In NBRD, we assume that the VIP takes random avoidance decisions according to alerts issued by a navigation-assistive system that lacks local path planning. We also assume that the VIP can move in a forward direction from $45°$ to $135°$ to manoeuvre the CMO with a limited distance (*i.e.,* $< 10m$). The second method is the state-of-the-art robot local path planning algorithm, improved *Artificial Potential Field* (APF) [16].

The APF is widely used in robotics motion planning due to its simplicity, effectiveness, and real-time performance. We chose the improved APF in [16] for comparison because the oscillation and local minima issues are effectively addressed by integrating the Simulated Annealing algorithm. To obtain comparable results, we set the parameter settings of improved APF, after extensive trials for parameter tuning while considering APF parameter settings effects studied in [17], as follows: repulsive coefficient $k_{rep} = 10^9$, gravitational coefficient $k_{att} = 0.1$, control annealing temperature $T = 10$ and escape step size $E_s = 4$, and repulsion influence distance $r_r = s_l(|v_O^{t_1}| + |v_B^{t_1}|)$. The value of r_r is



(a) (b) (c) (d)

Figure 3. The Distance Travelled by the VIP for Each CMO Class. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians.



(a) (b) (c) (d)

Figure 4. The Time Separation Between the VIP and Each CMO Class. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians.

adaptive based on the VIP's speed, CMO's speed, and the value of s_l. Note, s_l is set according to CMO's risk level, as the case of our algorithm's parameter settings, see Section 4.1. It should be noted that we set k_{rep} very large compared to k_{att} because our top priority is to ensure VIP's safety; we strive to keep the VIP as far away from threats as possible. We set a large value of E_s compared to [16] to speed up the processing of escaping local minima and oscillation. In our experiment, we do not care about the whole path of arriving at the destination planned by APF; we only consider the proposed path to avoid the CMO.

4.4. Results and Discussions

Travelling Distance. In Fig. 3, we show the result of the travelling distance of *MinD*, NBRD and improved APF [16] for each CMO class under 8 different moving speeds. Fig.3a shows the average results of 800 samples of the car class moving at speeds from $11km/h$ to $108km/h$, where 100 samples are averaged for each speed. In Fig. 3b, the results of the motorcycle class are demonstrated, with the same number of samples moving at the same speeds as the car class. In Fig. 3c and Fig. 3d, the results of bicycles and pedestrians moving at speeds of $7km/h \sim 54km/h$ and $3.6km/h \sim 36km/h$ are shown, respectively.

The results clearly demonstrate that the travelling distance of *MinD* scheme is minimised differently based on CMO class, the average distance is $2.91m$ for cars, $1.56m$ for motorcycles, $1.26m$ for bicycles, and $1.02m$ for pedestrians, and theses distances are fixed regardless of the CMO's speed for all classes. For NBRD, distances change randomly regardless of CMO class and speed, as they are not taken into account; the VIP moves as long as the alerts are sent until the danger is completely avoided. The average distance is $11.27m$ for cars, $8.79m$ for motorcycles, 7.91 for bicycles, and $7.62m$ for pedestrians, which are greater than our scheme's average distances. For APF, the travelling distance generally increases as CMO's speed increases for all CMO classes. This increase is because APF depends on two artificial forces for path planning, the attraction force to reach the destination and the repulsion force to move away from obstacles. The repulsion force is restricted by an obstacle's influence distance r_r, which becomes larger as the CMO's speed increases; this force becomes stronger at the centre of the r_r and decreases at its edges. Therefore, APF lets the VIP travel towards the destination (and towards the obstacle if they are in the same direction) until the repulsion force has a significant effect that keeps the VIP from hitting the obstacle. Based on this, the paths produced by APF have very long average distances, $21.58m$ for cars, $21.29m$ for motorcycles, $14.68m$ for bicycles and $11.53m$ for pedestrians. This makes APF impractical for VIPs as they have limited mobility constraints. To sum up, considering the minimum travel distance to avoid CMOs, *MinD* manages to achieve the lowest average distance over all classes, $1.69m$, compared to NBRD and improved APF, which achieved $8.90m$ and $17.27m$ on average, respectively. This means, with *MinD*, the VIP needs only $1.22s \sim 4.05s$ on average to avoid a threat when moving at $1.5km/h \sim 5km/h$, while with NBRD and improved APF, the VIP needs a longer time ($> 6.40s$ for NBRD and $> 12.43s$ for APF), which may threaten VIP's safety. This demonstrates the effectiveness of our scheme.

Time Separation. Fig. 4 shows the results of the time separation of *MinD*, NBRD and APF [16] for all CMO

Table 1. Collision Rate Among Our Scheme and Other Methods

Class	MinD (ours)	NBRD	Imp. APF[16]
Car	0%	2.75%	99.75%
Motorcycle	0%	3.38%	94.88%
Bicycle	0%	5.63%	84.38%
Pedestrian	0%	11.25%	68.75%
Average	0%	5.75%	86.94%

Table 2. Average Computation Time and Convergence Iteration

Method	Comp. Time	Convergance Iter.
MinD (ours)	0.04s	3
Imp. APF[16]	0.004s	60

classes. The results of cars and motorcycles are demonstrated in Fig. 4a and Fig. 4b, respectively. Fig. 4c shows the results of bicycles, while Fig. 4d shows the results of pedestrians. The results show that, for *MinD*, the time separation between the VIP and CMOs generally increases as speed increases; however, it decreases for cars and motorcycles at speeds $> 72km/h$. This is reasonable due to the considerably higher speeds of cars or motorbikes compared to the VIP, who possesses limited mobility. Despite this decrease, MinD still achieves large average time separations for such CMOs, i.e., $11.52s$ for cars, $11.59s$ for motorcycles, $8.69s$ for bicycles and $6.68s$ for pedestrians. NBRD shows a pattern similar to MinD but with minor fluctuations due to the randomness of the VIP's avoidance decisions (based on triggered alarms). It achieves $10.42s$ for cars, $10.53s$ for motorcycles, $7.41s$ for bicycles, and $5.20s$ for pedestrians. For APF, the time separation generally decreases as the speed increases for all CMOs, achieving $4.76s$ on average for cars, $4.84s$ for motorcycles, $4.12s$ for bicycles and $3.37s$ for pedestrians. The reason is that, in APF, the repulsion force generated by obstacles is inversely proportional to the distance. When the object becomes slower, it is detected closer to the VIP; thus, the repulsion force increases and vice versa. The APF operates contrary to our goal of providing more time separation as the CMO's speed increases. Therefore, we conclude that APF is unsafe for VIP local path planning.

Collision Rate. Table 1 compares collision rates for *MinD*, NBRD, and APF [16]. The results show that *MinD* achieves a 0% average collision rate over all classes, while NBRD and APF achieve 5.75% and 86.94% average collision rates, respectively. Unlike NBRD and APF, MinD prevented collisions because it adaptively optimises the VIP's position to avoid threats while considering the VIP's limited capabilities, the CMOs' moving patterns and the estimated collision times. This demonstrates the importance of our proposed scheme for planning collision-free routes for VIPs navigating dynamic environments with different CMOs moving at varying speeds.

Computation Time. Table 2 shows the comparison of *MinD* and APF [16] in terms of the average computation time and convergence iteration over all CMO classes. NBRD is not included here as it is not actual local path planning. Despite APF's significantly shorter computation time of $0.004s$, *MinD* successfully achieves real-time performance by only requiring $0.04s$ (close to $0s$), running on an Intel Core i9-9900K CPU. Due to the differences in the underlying avoidance strategies of both algorithms, AFP requires 60 iterations to converge, while MinD only requires 3 iterations. Unlike APF, MinD simultaneously considers the VIP's walking speed, CMOs' movement patterns, and estimated collision times and performs a look-ahead safety prediction of an optimal position; this accelerates convergence to a collision-free path that effectively accommodates the VIP in challenging environments. All these are done within only $0.04s$, demonstrating our algorithm's efficiency.

In summary, our proposed local path planning scheme demonstrates its effectiveness over the state-of-the-art local path planning APF approach and the baseline NBRD method for practical and safe VIP mobility assistance in terms of minimisation of CMO avoidance distance, significant time separation, and 0% collision rate, with a negligible processing speed. However, the *MinD* needs to be evaluated with CMOs that unexpectedly change their trajectory, which is our future step. Additionally, we plan to investigate the *MinD* performance with more complex real-world scenarios and when it is fully integrated with a vision-based system under different conditions (*e.g.,* lighting conditions). We also will optimise the MinD to achieve real-time performance on resource-constrained hardware.

5. Conclusion

This paper presents *MinD*, a VIP-specific local path planning scheme to assist VIPs in outdoor mobility by optimally avoiding CMOs moving at different speeds. In formulating the *MinD*, we explored the natural human behaviour to avoid moving threats considering VIP's conditions and perceptions in practice. We also conduct a look-ahead safety prediction trajectory analysis of the optimal path to guarantee more VIPs' safety. Experimental results demonstrate the effectiveness of *MinD*, compared to the well-known APF algorithm, in terms of minimising the travelling avoidance distance (only $1.69m$ on average over all CMO classes, which is shorter than APF by 90.23%), allowing a large time separation between the VIP and CMO ($> 5.35s$, on average, compared to APF), and avoiding the threats by 100%, with near real-time performance where the processing time is $0.04s$ on a standard off-the-shelf laptop.

References

[1] Chapter 10 - trajectory planning of tractor semitrailers. In *Vehicle Dynamics and Control*, pages 429–478. Elsevier, 2021.

[2] Andreas Antoniou and Wu-Sheng Lu. *Practical optimization*. Springer, New York, NY, 1 edition, 2007.

[3] Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, and Dijun Liu. Virtual-blind-road following-based wearable navigation device for blind people. *IEEE Transactions on Consumer Electronics*, 64(1):136–143, 2018.

[4] K. Cai, C. Wang, J. Cheng, C. W. de Silva, and M. Q.-H. Meng. Mobile robot path planning in dynamic environments: A survey. *CoRR*, abs/2006.14195, 2020.

[5] Qiaoyu Chen, Lijun Wu, Zhicong Chen, Peijie Lin, Shuying Cheng, and Zhenhui Wu. Smartphone based outdoor navigation and obstacle avoidance system for the visually impaired. In Rapeeporn Chamchong and Kok Wai Wong, editors, *Multi-disciplinary Trends in Artificial Intelligence*, pages 26–37. Springer, 2019.

[6] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE, 2017.

[7] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 285–292. IEEE, 2017.

[8] Edson Soares da Silva, Gabriela Fischer, Rodrigo Gomes da Rosa, Pedro Schons, Luísa Beatriz Trevisan Teixeira, Wouter Hoogkamer, and Leonardo Alexandre Peyré-Tartaruga. Gait and functionality of individuals with visual impairment who participate in sports. *Gait & Posture*, 62:355–358, 2018.

[9] Ping-Jung Duh, Yu-Cheng Sung, Liang-Yu Fan Chiang, Yung-Ju Chang, and Kuan-Wen Chen. V-eye: A vision-based navigation system for the visually impaired. *IEEE Transactions on Multimedia*, 23:1567–1580, 2021.

[10] Fatma El-Zahraa El-Taher, Ayman Taha, Jane Courtney, and Susan Mckeever. A systematic review of urban navigation systems for visually impaired people. *Sensors*, 21(9):1–35, 2021.

[11] Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3052–3059. IEEE, 2018.

[12] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[13] Stephen J Guy, Jatin Chhugani, Changkyu Kim, Nadathur Satish, Ming Lin, Dinesh Manocha, and Pradeep Dubey. Clearpath: Highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187. Association for Computing Machinery, 2009.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, University Printing House, Cambridge CB2 8BS, United Kingdom, 2 edition, Dec. 2004.

[15] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Gräser. Disnet: a novel method for distance estimation from monocular camera. In *Proceedings of the 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, pages 1–6, 2018.

[16] Naifeng He, Yifan Su, Xiaoliang Fan, Zihong Liu, Bolun Wang, et al. Dynamic path planning of mobile robot based on artificial potential field. In *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pages 259–264. IEEE, 2020.

[17] Farrokh Janabi-Sharifi and David Vinke. Integration of the artificial potential field approach with simulated annealing for robot path planning. In *Proceedings of 8th IEEE International Symposium on Intelligent Control*, pages 536–541, 1993.

[18] Pengfei Lin, Woo Young Choi, and Chung Choo Chung. Local path planning using artificial potential field for waypoint tracking with collision avoidance. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020.

[19] Yimin Lin, Kai Wang, Wanxin Yi, and Shiguo Lian. Deep learning based wearable assistive system for visually impaired people. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2549–2557, 2019.

[20] Payal TusharKumar Mahida, Seyed Shahrestani, and Hon Cheung. Dynapath: Dynamic learning based indoor navigation for vip in iot based environments. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 8–13. IEEE, 2018.

[21] N. Sathya Mala, S. Sushmi Thushara, and Sankari Subbiah. Navigation gadget for visually impaired based on iot. In *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, pages 334–338. IEEE, 2017.

[22] Abir Benabid Najjar, Arwa Rashed Al-Issa, and Manar Hosny. Dynamic indoor path planning for the visually impaired. *Journal of King Saud University-Computer and Information Sciences*, pages 1–11, 2022.

[23] Durgesh Nandini and KR Seeja. A novel path planning algorithm for visually impaired people. *Journal of King Saud University-Computer and Information Sciences*, 31(3):385–391, 2019.

[24] Waqas Nawaz, Kifayat Ullah Khan, and Khalid Bashir. A review on path selection and navigation approaches towards an assisted mobility of visually impaired people. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(8):3270–3294, 2020.

[25] Ippei Nishitani, Tetsuya Matsumura, Mayumi Ozawa, Ayanori Yorozu, and Masaki Takahashi. Human-centered x–y–t space path planning for mobile robot in dynamic environments. *Robotics and Autonomous Systems*, 66:18–26, 2015.

[26] Cao Qixin, Huang Yanwen, and Zhou Jingliang. An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3331–3336. IEEE, 2006.

[27] Samik Raychaudhuri. Introduction to monte carlo simulation. In *2008 Winter Simulation Conference*, pages 91–100, 2008.

[28] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767:1–6, Apr. 2018.

[29] Ruxandra Tapu, Bogdan Mocanu, and Titus Zaharia. Deepsee: Joint object detection, tracking and recognition with application to visually impaired navigational assistance. *Sensors*, 17(11):1–24, 2017.

[30] Tolgahan Turker, Ozgur Koray Sahingoz, and Guray Yilmaz. 2d path planning for uavs in radar threatening environment using simulated annealing algorithm. In *2015 international conference on unmanned aircraft systems (ICUAS)*, pages 56–61. IEEE, 2015.

[31] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008.

[32] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402:1–5, Mar. 2017.

[33] Akihiro Yamashita, Kei Sato, and Katsushi Matsubayashi. Walking navigation system for visually impaired people based on high-accuracy positioning using qzss and rfid and obstacle avoidance using hololens. *International Journal of Innovative Computing, Information and Control*, 16(4):1459–1467, 2020.

[34] Taufik S. T. Yusof, Siti Fauziah Toha, and Hazlina Md. Yusof. Path planning for visually impaired people in an unfamiliar environment using particle swarm optimization. *Procedia Computer Science*, 76:80–86, 2015.