2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

GPF-BG: A Hierarchical Vision-Based Planning Framework for Safe Quadrupedal Navigation

Shiyu Feng[1], Ziyi Zhou[1], Justin S. Smith[2], Max Asselmeier[1], Ye Zhao[1] and Patricio A. Vela[2]

***Abstract*— Safe quadrupedal navigation through unknown environments is a challenging problem. This paper proposes a hierarchical vision-based planning framework (GPF-BG) integrating our previous Global Path Follower (GPF) navigation system and a gap-based local planner using Bézier curves, so called *Bézier Gap* (BG). This BG-based trajectory synthesis can generate smooth trajectories and guarantee safety for point-mass robots. With a gap analysis extension based on non-point, rectangular geometry, safety is guaranteed for an idealized quadrupedal motion model and significantly improved for an actual quadrupedal robot model. Stabilized perception space improves performance under oscillatory internal body motions that impact sensing. Simulation-based and real experiments under different benchmarking configurations test safe navigation performance. GPF-BG has the best safety outcomes across all experiments.**

I. Introduction

Quadrupedal robots have demonstrated superior terrain traversability compared to traditional wheeled robots [1]–[3]. Significant progress has been made during the past few years to improve the robustness and agility of legged locomotion control [4]–[10], which enables the incorporation of exteroceptive sensors for autonomous navigation (e.g. Fig. 1). Taking into account legged robot morphology, prior navigation works have been mostly focused on rough terrain traversability [2], [11]–[15], multi-modal planning [2], [16]–[19], and multi-robot exploration [20], [21]. However, navigation safety and obstacle avoidance have not been formally analyzed, and the tested scenarios are limited to less dense environments. Safe navigation through unknown or partially unknown environments for quadrupeds is critical to the field deployment of legged robots yet remains under explored.

Vision-based navigation using depth sensing provides rich perceptual information and distance estimates to objects, which subsequently supports terrain traversal and obstacle avoidance by mobile robots. Further improvements arise from a hierarchical structure whereby hierarchical problem decompositions support safe navigation assertions [22]. The global path follower (GPF-X) navigation system [23] in our previous work generates safe paths and additionally checks collisions through Planning in Perception Space (PiPS) [24] to guarantee safety. A Potential Gap [22] local planner is guaranteed to be safe for holonomic point-mass robots. Integrating this local planner with GPF-X can theoretically prove safety. However, this framework is only designed for wheeled mobile robots. Generalizing it to quadrupedal robots needs further effort in a principled manner.

Fig. 1: Unitree A1 quadruped navigates an unknown, constrained environment. Depicts robot navigation paths for GPF-BG (green) and TEB (red).

This paper proposes a novel hierarchical vision-based system, GPF-BG (in Fig. 2), for safe quadrupedal navigation. BG stands for *Bézier Gap* local planner, which is an improved version of Potential Gap. It incorporates robot orientation and geometry during gap detection and trajectory synthesis based on quadrupedal motion affordances. Further, due to robot body oscillations during dynamic locomotion, a perception space stabilization mechanism is proposed and integrated for correcting egocentric internal motion artifacts in perception space.

The contributions of this paper are enumerated below and annotated in Fig. 2. Quadrupedal navigation relies on the GPF-X hierarchical navigation system as the basis for safe navigation (Section III). **(i)** The proposed *Bézier Gap* local planner (Section IV-A) offers a new Bézier-based trajectory synthesis method for gaps. Safety is guaranteed for point-mass robots and translated to rectangular-body-shape quadrupeds using a robot geometry extension module. **(ii)** Stabilized perception space provides safer collision checking when the locomotion dynamics induce large egocentric internal motion (Section IV-B). **(iii)** Simulation benchmark and real experiments demonstrate the vision-based planning framework for safe quadrupedal navigation (Section V).

II. Related Work

Quadrupedal Navigation: Previous works have achieved impressive results integrating navigation and legged locomotion control [2], [11]–[20]. Specifically, medium and large-size quadrupeds in [11], [13], [14] used laser scanners to detect obstacles for navigation and environment exploration [20]. Stereo camera usage was limited in terrain mapping

[1]S. Feng, Z. Zhou, M. Asselmeier, and Y. Zhao are with the School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA. shiyufeng@gatech.edu

[2]J. Smith and P.A. Vela are with the School of Electrical and Computer Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30308, USA.

*This work supported in part by NSF Award #1849333, #2144309 and Georgia Tech IRIM/IPaT Aware Home Seed Grant.

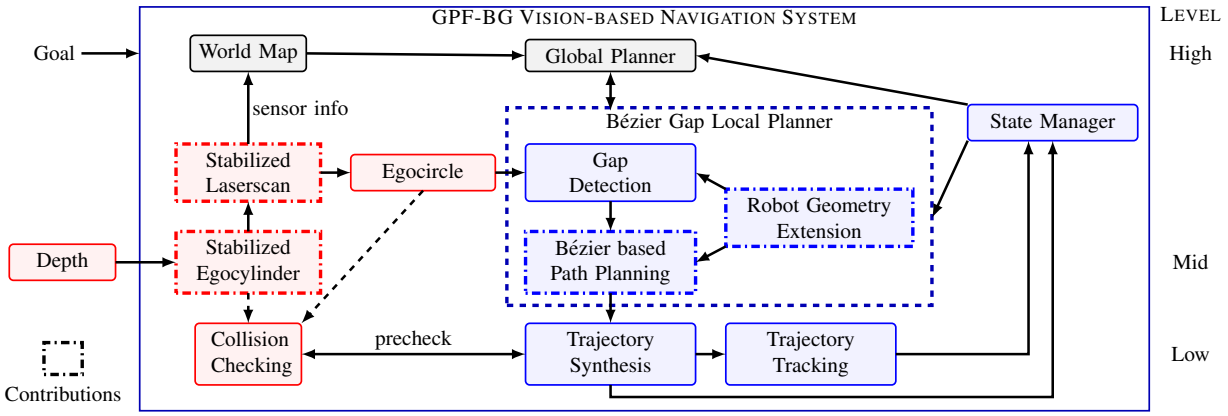


Fig. 2: GPF-BG hierarchical navigation system. The red blocks include perception space. Blue blocks present following and tracking modules. The dashed arrows indicate the option to choose either egocylinder or egocircle for collision checking.

[13], [14]. In [2], [15], [18], [19], stereo and tracking cameras were used on small-scale quadrupeds for terrain mapping, navigation, and localization. Off-the-shelf or customized path planners were used to allow collision avoidance or multi-modal selection. However, none of them formally studied safety for vision-based navigation.

Safety in Vision-based Navigation: Hierarchical systems have long been applied to robotic platforms due to their increased robustness and fault tolerance [25]–[30]. Thus, safety across the hierarchical levels will be reviewed.

At the planning level, both global and local planners can find collision-free paths. Graph search methods such as Dijkstra's algorithm, A*, D*, and sampling-based methods, e.g. Probabilistic Road Map (PRM) and Rapidly-exploring random trees (RRT) perform global planning in discrete spaces [24]. Local reactive planners, e.g. Dynamic Window Approach (DWA) [31] and Timed-Elastic-Bands (TEB) [32], generate safe velocity commands based on costmap or optimal trajectories. Some gap-based local methods [22], [33]–[38] use the gap as navigation affordance and incorporate sensory data into planning. Potential Gap [22] extends them with safety guarantees for ideal robots. It provides a theoretical foundation for safe quadrupedal navigation.

Safety in control avoids obstacles in the low-level parts of the hierarchy. A real-time vision-based navigation method based on Hamilton-Jacobi reachability can guarantee strong safety in unknown environments [39]. Control barrier functions (CBF) also serve as constraints applied on the control to reach collision-free sets [40].

Moreover, fast collision checking assures safety. PiPS [24] collision checking with egocylinder is fast, and propagates temporal sensing information to improve safety with a limited sensor field of view. Stixel egocentric navigation [23] is an alternative sparse method using passive stereo vision with good scalability across different computational platforms.

Trajectory Synthesis from Potential Field: The Potential Gap local planner [22] synthesizes trajectories based on potential methods. Robot orientation and dynamics are not considered, which may generate local trajectories with high tracking error, thereby resulting in collisions. Bézier curves can address this, as they are known to synthesize smooth, short-time trajectories for various robot models [41]–[46], which can improve navigation safety for quadrupedal robots.

III. Hierarchical Navigation System

The GPF-X hierarchical navigation system [23] serves as the framework for defining a quadrupedal navigation method, whose system flowchart is given in Fig. 2. GPF-X maintains safety through the hierarchical structure. In the global following mode, the system tracks the presumed collision-free trajectory synthesized from the global plan. If the sensing indicates an invalid trajectory segment, the state manager switches to trajectory based local planners. The synthesized local trajectories consider safety and passibility. A gap-based local planner, called *Bézier Gap* (BG), is proposed in §IV-A to guarantee safety for point mass robot. It is an improved version of Potential Gap (PG) [22] by redesigning path synthesis. An extension for quadrupedal robot geometry is added to balance traversability and safety. The entire system will be abbreviated as GPF-BG. PiPS [24] collision checking is performed on both global and local trajectories to prevent collision. Due to potential internal movements during quadrupedal locomotion, a stabilized perception space is designed for successful planning with the egocircle and collision checking with the egocylinder. The result is safe quadrupedal navigation achieved by the hierarchical system.

IV. Bézier Gap Local Planning

This section describes the design of a *Bézier Gap* local planner including trajectory synthesis, safety guarantee, and geometry extension. It also covers stabilized perception space for improved egocylinder and egocircle modeling.

A. Bézier Gap Local Planner

The Potential Gap local planner [22] utilizes gaps as navigation affordances. Gaps offer fast free space modeling from egocentric perception based on line-of-sight visibility. Safety is guaranteed for holonomic point-mass robots. However, due to the dynamic constraints and geometry of quadrupedal robots, the Potential Gap method cannot maintain safe navigation. Modifications to trajectory synthesis should be made to improve safety. The gap set $\mathcal{G}$ as detected from the egocircle $\mathcal{L}$ used in Potential Gap remains in Bézier Gap.

1) Bézier-based Trajectory Synthesis: In Potential Gap, the path is integrated based on the vector field constructed from gaps without any consideration of robot orientation and dynamics. Switching to a new local path may lead to sharp turns that are dangerous to quadrupedal robots. Smoother trajectory generation is required. Quadratic Bézier curves can create smooth paths satisfying initial orientation and terminal way point constraints. The closed form of quadratic Bézier curve is parameterized by u,

$$\boldsymbol{B}(u) = (1-u)^2\boldsymbol{Q}_0 + 2(1-u)u\boldsymbol{Q}_1 + u^2\boldsymbol{Q}_2, 0 \leq u \leq 1 \quad (1)$$

where $\boldsymbol{Q}_i$ is the control point. For each gap $G \in \mathcal{G}$, the robot initial position $\boldsymbol{p}(0)$ is used as the first control point $\boldsymbol{Q}_0 = \boldsymbol{p}(0)$. The local way point position $\boldsymbol{p}_{\text{wpt}}$ is set as the last control point $\boldsymbol{Q}_2 = \boldsymbol{p}_{\text{wpt}}$. The second control point is constructed based on robot initial orientation $\theta(0)$ and forward linear velocity $\nu(0)$. The unit orientation vector is $\vec{\boldsymbol{o}}(0) = [cos(\theta(0)), sin(\theta(0))]$. The vector $\boldsymbol{Q}_1 - \boldsymbol{Q}_0$ should be co-linear with the orientation vector $\vec{\boldsymbol{o}}$. Curve velocities are obtained from the derivative of quadratic Bézier curve,

$$\boldsymbol{B}'(u) = 2(1-u)(\boldsymbol{Q}_1 - \boldsymbol{Q}_0) + 2u(\boldsymbol{Q}_2 - \boldsymbol{Q}_1) \quad (2)$$

where $\boldsymbol{B}'(0) = 2(\boldsymbol{Q}_1 - \boldsymbol{Q}_0)$. Set $\boldsymbol{B}'(0) = \nu(0)$, which requires $||\boldsymbol{Q}_1 - \boldsymbol{Q}_0|| = \nu(0)/2$. The position, orientation, and speed constraints uniquely define the control points:

$$\boldsymbol{Q}_0 = \boldsymbol{p}(0), \quad \boldsymbol{Q}_1 = \boldsymbol{p}(0) + \frac{\nu(0)}{2}\vec{\boldsymbol{o}}(0), \quad \boldsymbol{Q}_2 = \boldsymbol{p}_{\text{wpt}}. \quad (3)$$

The synthesized trajectory synthesized valid for $u \in [0, 1]$ is

$$\mathcal{T}(u) = (1-u)^2\boldsymbol{p}(0) + (1-u)u[2\boldsymbol{p}(0) + \nu(0)\vec{\boldsymbol{o}}(0)] + u^2\boldsymbol{p}_{\text{wpt}}. \quad (4)$$

which is parameterized by u instead of time t.

2) Trajectory Reparameterization: A second pass reparameterizes the raw trajectory based on desired linear velocity ν_d. The desired unit length is $l_d = \nu_d(t_{i+1} - t_i)$. The new trajectory is represented as a set of poses $\mathcal{T}(t) = \{\boldsymbol{p}(t_i) \mid 0 \leq t_i \leq t_{\text{end}}\}$ with consecutive poses having length l_d.

3) Safety Guarantee for Point-mass Robot: The safety can be guaranteed through the design of Bézier trajectory synthesis. A gap example is shown in Fig. 3. Two gap lines, L_l and L_r, are represented as blue lines starting from the robot origin. The gap lines construct a gap triangle G_{tri}. The red circle G_{circ} is the largest circular free space in egocircle. The entire free space $\mathcal{F}$ is the union of gap triangle and circle $\mathcal{F} = G_{tri} \cup G_{circ}$. The region outside the gap arc is also included in this free space. Moreover, the convex hull of Bézier polygon $\mathcal{B}$ contains Bézier curve. If $\mathcal{B} \subseteq \mathcal{F}$, the safety is guaranteed for point-mass robot when following the Bézier trajectory.

The safety guarantee problem becomes the design of three control points. The first and second control points are found through (3). Assuming the second control point is within gap circle $\boldsymbol{Q}_1 \in G_{circ}$, the local way point is a free point to keep safety. Define the intersection points of gap lines and circle as $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$. The line constructed by second control point $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_l$ is denoted as $L_{1,l}$. The line created by first and second control points is $L_{0,1}$. A condensed safe region $\mathcal{F}_s \subset \mathcal{F}$ is enclosed by L_l, $L_{0,1}$, $L_{1,l}$ and all regions outside the gap arc. The safety is guaranteed by placing the local way point $\boldsymbol{p}_{\text{wpt}}$ in the condensed free region.

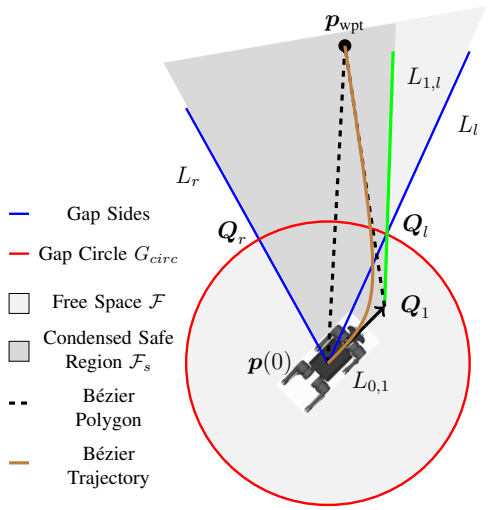


Fig. 3: Bézier trajectory synthesis. $\boldsymbol{p}_0$ is the robot origin. $\boldsymbol{Q}_1$ is the second control point. Blue lines L_l and L_r are left and right gap lines. Red circle is the largest circular free space in egocircle. $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$ are the left and right intersection points of G_{circ} and G_{tri}. $L_{0,1}$ is the line segment constructed by $\boldsymbol{p}_0$ and $\boldsymbol{Q}_1$, and also represents the robot orientation. $L_{1,l}$ is the line through $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_l$. Local way point $\boldsymbol{p}_{\text{wpt}}$ is inside the condensed safe region $\mathcal{F}_s$ to guarantee safety. Dashed lines show the Bézier polygon. Brown is the synthesized trajectory.

$$\boldsymbol{p}_{\text{wpt}} \in \mathcal{F}_s \Rightarrow \mathcal{B} \subseteq \mathcal{F} \Rightarrow \text{Safety Guaranteed} \quad (5)$$

The candidate $\boldsymbol{p}_{\text{wpt}}$ is found from raw global plan. Of note, if the second control point is outside the gap circle, this gap will be ignored for generating trajectories. When all second control points are not within gap circle, a recovery behavior is triggered by rotating robots.

4) Quadrupedal Robot Geometry Extension: Since safety is only guaranteed for point-mass robots, quadrupedal robot geometry should be considered during gap analysis and path planning. When considering the robot geometry, two boundary lines L_r and $L_{1,l}$ of the condensed safe region $\mathcal{F}_s$ will be rotated inwards to prevent a collision from $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$. The safety buffer is a function of robot's body width. The Potential Gap local planner normally assumes circular mobile robots so that a constant robot radius suffices for the safe buffer value. However, when applying to a quadrupedal robot, this assumption does not hold any more, due to the fact that using the robot's diagonal length as a constant parameter is overly conservative. The path planning may fail when the robot can actually pass. Safe gap passage determination should depend on quadrupedal robot orientation. Therefore, two equivalent lengths are used to comprise the gap analysis.

Equivalent Passing Length l_{ep}. As shown in Fig. 4(a), the equivalent passing length l_{ep} is denoted by the distance between two lines that are parallel to gap direction and maximally enclose the rectangle. It assesses the passibility of gaps. If the gap length $l_{gap} > l_{ep}$, the gap is labelled as passable. This equivalent value is a function of the angle

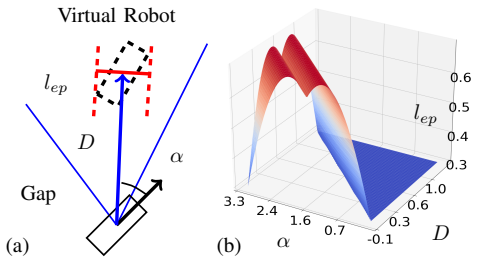


Fig. 4: Equivalent passing length. (a) l_{ep} finding. Black arrow is robot orientation. Blue arrow is gap direction. α is the angle difference between robot orientation and gap direction. D is the distance to the virtual pose. Red line l_{ep} is normal to gap direction. (b) An example of $l_{ep}(\alpha, D)$. The robot geometry is 0.7m x 0.3m. $\nu_d = 0.2$m/s, $\omega_d = 0.5$rad/s. It shows the dependence on robot orientation.

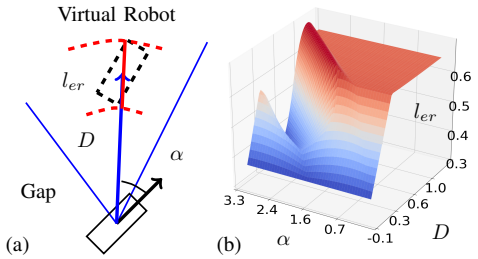


Fig. 5: Equivalent radial length. (a) The red solid line is l_{er}. Two red dashed curves are the largest and smallest circle arcs that can enclose virtual robot boundary. (b) An example of $l_{er}(\alpha, D)$.

α between gap direction and robot orientation. However, in the planning process, forward directed gap passage is always preferred, i.e. eventually aligning robot orientation with gap direction. But due to the robot speed $\boldsymbol{v}_d = [\nu_d, \omega_d]$, the alignment is gradually achieved. The actual l_{ep} should be calculated based on the future robot pose passing through the gap. A virtual robot is placed to represent this future pose. Its orientation can be estimated from the distance D to the virtual pose and $\boldsymbol{v}_d$. Assume α is positive.

$$\alpha(D) = \begin{cases} \alpha(0) - \omega_d \frac{D}{\nu_d}, & \alpha(D) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

l_{ep} becomes a function of α and D. An example of $l_{ep}(\alpha, D)$ is shown in Fig. 4(b). If D is large enough, the alignment is always achievable. l_{ep} becomes the robot width.

Equivalent Radial Length l_{er}. Similar to l_{ep}, an example demonstrating the equivalent radial length l_{er} is shown in Fig. 5(a). A virtual robot will be placed at the future pose. This equivalent value measures the radial difference between the largest and smallest circles that have intersections with the virtual robot boundary. It is used in radial gap manipulation and as the buffer size when radially shrinking the egocircle. Same as l_{ep}, this equivalent length is the function $l_{er}(\alpha, D)$, as shown in Fig. 5(b). When D is large enough, l_{er} becomes a value that is slightly larger than robot length.

5) Trajectory Scoring: The Bézier-based trajectory is synthesized for each gap. A set of raw trajectories will be scored to pick the best for following. The scoring function is similar to Potential Gap [22] with necessary modifications for rectangular robot geometry.

$$J(\mathcal{T}) = \sum_{\boldsymbol{p} \in \mathcal{T}} C(d_{\text{bound}}(\boldsymbol{p}, \mathcal{L})) + w_1 ||\boldsymbol{p}_{\text{end}} - \boldsymbol{p}_{\text{goal}}||, \text{ where}$$

$$C(d_{\text{bound}}) = \begin{cases} c_{\text{obs}} e^{-w_2 d_{\text{bound}}}, & d_{\text{bound}} > 0 \\ 0, & d_{\text{bound}} > r_{\max} \\ \infty, \text{otherwise} \end{cases}$$

$d_{\text{bound}}(\boldsymbol{p}, \mathcal{L})$ is the nearest distance from robot boundary to egocircle at a trajectory pose $\boldsymbol{p}$. w_1, w_2, c_{obs} and $r_{\max}$ are tuning parameters and described in [22]. $\boldsymbol{p}_{\text{goal}}$ is the local goal found from global plans.

B. Stabilized Perception Space

Correct gap analysis and collision geometry requires correct sensor data interpretation in the face of potential egocentric internal motions of the body. Stabilized perception space compensates for this as needed.

1) Egocylinder: The egocylinder is an egocentric means of representing the local environment [24]. It consists of a virtual cylinder whose surface is a range image I_c. As a 2D, perception space representation, it benefits from low latency and computational cost. A point is added to the egocylinder by projecting it onto the virtual cylinder and storing the range of the point in the corresponding pixel of the range image. As the robot moves, the values in I_c are projected back into Cartesian coordinates, transformed by the robot's relative motion, and projected back onto I_c. By placing the egocylinder's origin at the same position as the sensor (eg. depth camera), the projective nature of the egocylinder ensures that each pixel of I_c contains the range of the nearest (known) object in that direction.

However, rigidly fixing the egocylinder's frame to that of the sensor can have undesirable consequences if the sensor is subjected to significant oscillation. Since the egocylinder can only store one point in each direction, updates frequently result in some points being lost as multiple points map to the same pixel. As a result, the loss of points is exacerbated by frequent oscillations around the pitch and roll axes. In this situation, it is beneficial to define a virtual, *stabilized* frame to compensate for some of the motion and reduce the point loss. While the position of this frame is still fixed to the sensor's, its orientation isn't. The orientation is defined such that the frame remains level relative to the global frame and only rotates around its yaw axis.

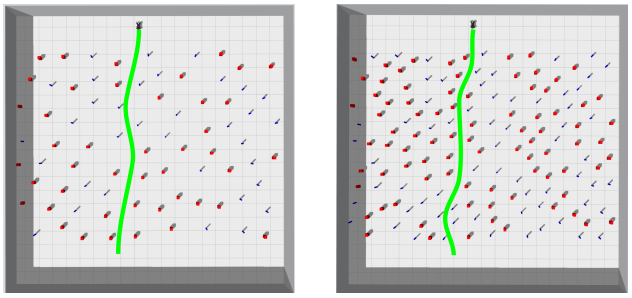
2) Egocylinder to laser scan: Another benefit of the stabilized egocylinder is that it simplifies the process of generating a 1D synthetic laser scan representation of the local environment. Upper and lower planes define the volume relative to the robot within which a point should be considered an obstacle. Projecting a ray through each row of I_c and calculating its intersection with either plane yields the range threshold below which a point represents an obstacle. Converting I_c to a laser scan is simply a matter of generating a mask of all pixels with values less than their rows' thresholds and applying a column-wise *min* reduction to the masked I_c.

V. Experiments

A. Simulation Configurations

Performance evaluation of GPF-BG uses the outcomes from Monte Carlo ROS/Gazebo simulation benchmarking, implemented on a dual Intel Xeon E5-2680 workstation (single-thread Passmark score: 1547).

1) Scenario setup: The scenarios from [22] are used: Sector, Dense, Campus, and Office worlds. Obstacles are randomly spawned in the last three environments. The minimum inter-obstacle distance controls the environment complexity. Smaller means denser; 1.5 m and 1 m are tested to assess the safe navigation capability of navigation frameworks. Fig. 6 shows different complexities of the dense world.



(a) 1.5m (b) 1m

Fig. 6: Dense worlds with different complexity with the robot at the top and actual paths (green) depicted. In the more complex environment, GPF-BG still achieves successful navigation with similar path length.

2) Robot configurations: Three different robot configurations are used. 1) STDR box robot *meeting the assumptions of GPF-BG for guaranteed safety*: a 2D planar robot. The sensor is a laser scanner with a 360-degree field of view (FoV). 2) RidgeDog Gazebo robot: a 3D robot with the appearance of Unitree A1 model *meeting the assumptions of GPF-BG for guaranteed safety*. The simulated holonomic mobile robot motion does not involve moving legs, replicating robot geometry but removing the robot oscillations due to locomotion. The sensor is a depth camera with a 60-degree FoV. A virtual laser scan is created from the depth camera for the costmap and egocircle. The egocylinder is used for collision checking without the need for a stabilized perception space. 3) Unitree A1 robot: a full-order model with 3 actuated joints on each leg and 18-DoF in total, *partially meeting the assumptions of GPF-BG*. The low-level controller from [7] is used to control the robot. Perception spaces, egocylinder and egocircle, are built based on the same depth camera with stabilization described in §IV-B.

3) Navigation systems: To compare with GPF-BG, we use other three navigation systems: move_base with DWA [31], move_base with TEB [32] and GPF-PG (GPF-X with Potential Gap local planner). The global and local costmaps in the above systems are created from pure laserscan, depth image to laserscan and stabilized laserscan.

The Monte Carlo benchmark provides statistics to evaluate performance. There are 80 runs per navigation system with STDR and RidgeDog, and 40 with A1. Success, abort and collision rates are recorded to compare results.

B. Results and Analysis

The results are in Tables I and II. STDR and RidgeDog do not need stabilized perception space. A1 needs perception stabilization for the robot oscillations. The comparison between GPF-BG and TEB are shown in Fig. 7.

1) STDR and RidgeDog: GPF-BG can achieve a 100% success rate without collision. For STDR, DWA and TEB do not have any collision but more abort cases than GPF-BG, since 360-degree FoV is used to sense the laser-safe environments. It indicates that GPF-BG not only can maintain safety, but also consider passibility through the environment.

For RidgeDog, there are a few collision cases in DWA and TEB due to the limited FoV. GPF-BG uses egocylinder and egocircle for collision checking and planning, which store and propagate temporal information to improve safety. By design, the BG local planner synthesizes visibility-constrained trajectories to reduce the effects of limited FoV.

When increasing the environment complexity from 1.5 m to 1 m, the success rates of DWA and TEB drop around 4%. GPF-BG has no reduction to have a 100% success rate.

2) A1: GPF-BG has a higher success rate and less collision rate than DWA and TEB. The collisions are caused by limited FoV and failed robot movement. The tested navigation systems only consider velocity constraints of the quadrupedal robot without incorporating full dynamics. GPF-BG has a lower success rate drop (only 2.5%) than DWA (10%) and TEB (7.5%) when testing in more complex environments. Fig. 7 shows the smaller performance drop and more consistent performance of GPF-BG (versus the baselines) caused by legged realization and changing environment complexities. Robot paths in the dense world with different complexity are shown in Fig. 6.

3) Comparison between BG and PG: GPF-PG A1 runs use stabilized perception space for fair comparison. GPF-PG success rate is better than or equal to DWA and TEB, but worse than GPF-BG. GPF-BG has perfect performance in the scenarios meeting the assumptions underlying GPF-BG, while GPF-PG does not. For the A1 robot, the GPF-PG collision rate is higher than GPF-BG by 5% in the hard case.

4) Comparison for stabilized perception space: In simulation, only A1 robot requires stabilized perception space. The results are in Table III. GPF-BG$^-$ denotes the GPF-BG navigation system without stabilized perception space. The collision rate increases by 12.5%.

5) Conclusions: From the simulation benchmarking, GPF-BG has the best safety performance among tested systems. It also has more consistent performance (lower variance across configuration changes). The analysis indicates that the *Bézier Gap* local planner and stabilized perception space improve quadrupedal navigation safety.

C. Quadrupedal Navigation Evaluation

1) Experiment setup: The Unitree A1 is equipped with an Intel D435i depth camera and a 2D laser scanner. The laser scanner is only used for a built-in SLAM package generating a static global map without the obstacles discussed below, as well as providing the odometry data. The stabilized

TABLE I: Simulation Benchmark (minimum distance between obstacles = 1.5m)

STDR	Success	Abort	Collision
DWA	100%	0%	0%
TEB	97.5%	2.5%	0%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

RidgeDog	Success	Abort	Collision
DWA	98.75%	0%	1.25%
TEB	100%	0%	0%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

A1	Success	Abort	Collision
DWA	85%	0%	15%
TEB	85%	2.5%	12.5%
GPF-PG	85%	0%	15%
GPF-BG	87.5%	0%	12.5%

TABLE II: Simulation Benchmark (minimum distance between obstacle = 1m)

STDR	Success	Abort	Collision
DWA	96.25%	3.75%	0%
TEB	93.75%	6.25%	0%
GPF-PG	96.25%	1.25%	2.5%
GPF-BG	100%	0%	0%

RidgeDog	Success	Abort	Collision
DWA	95%	2.5%	2.5%
TEB	96.25%	2.5%	1.25%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

A1	Success	Abort	Collision
DWA	75%	10%	15%
TEB	77.5%	0%	22.5%
GPF-PG	80%	0%	20%
GPF-BG	85%	0%	15%

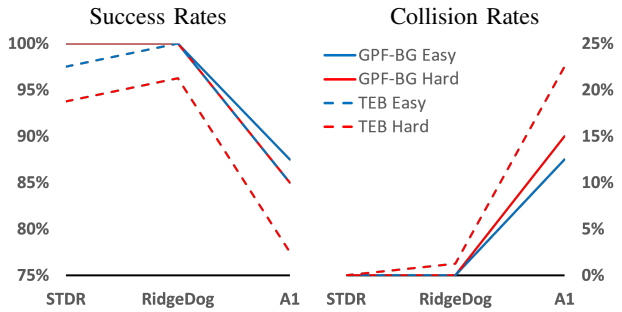


Fig. 7: Simulation comparison between GPF-BG and TEB in easy and hard scenarios for three robot models. GPF-BG has higher success rates and lower collision rates overall than TEB. The performance drops from rolling to A1 legged robot model and environment complexities are smaller.

TABLE III: Comparison of Stabilized Perception Space

A1	Success	Abort	Collision
GPF-BG	85%	0%	15%
GPF-BG$^-$	72.5%	0%	27.5%

TABLE IV: Real Experiment Results

	Success		Abort		Collision	
	Easy	**Hard**	**Easy**	**Hard**	**Easy**	**Hard**
DWA	1/5	0/5	4/5	5/5	0/5	0/5
TEB	4/5	2/5	1/5	1/5	0/5	2/5
GPF-BG	5/5	4/5	0/5	0/5	0/5	1/5

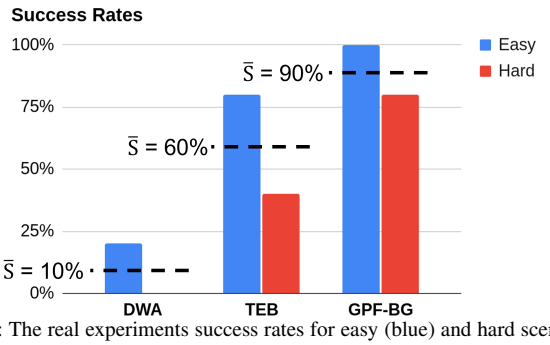


Fig. 8: The real experiments success rates for easy (blue) and hard scenarios (red). Annotations are average success rates.

perception space is disabled in hardware tests, since the A1 moves more stably than simulation. However, it can drastically improve navigation performance on quadrupedal robots with more oscillations and higher speed.

An indoor scenario is set up in Fig. 1. The courses are characterized into easy and hard ones depending on the minimum distance between the obstacles. The easy course consists of three garbage cans (two smaller blue cans and one larger green can) with a 2m minimum obstacle distance. The hard course keeps the same setup but with an additional chair that further shrinks the minimum distance to 1.2m. The start and end poses are same among all experiments. 5 repeated runs are tested for each navigation system.

2) Experiment results and analysis: The results of real experiment are in Table IV. DWA planner cannot perform well with the real quadrupedal robot. In both easy and hard scenarios, GPF-BG has a higher success rate and less collision than TEB. The collisions of GPF-BG are also caused by limited camera FoV and quadrupedal locomotion which are similar to the simulation benchmark. In the more complicated environment, GPF-BG has a lower success rate performance drop as shown in Fig. 8, which is consistent with simulation results. Fig. 1 depicts the robot's actual paths for GPF-BG and TEB in the hard environment testing. GPF-BG safely reaches the goal with a shorter path than TEB. These real experiments further demonstrate the safer navigation performance of quadrupedal robots.

The computation of GPF-BG has a bimodal property [23]. The following stage takes < 5 ms. The planning mode is real time and takes ~ 65 ms on average.

VI. Conclusion

The proposed GPF-BG is a hierarchical vision-based planning framework for safe quadrupedal navigation. *Bézier Gap* local planner translates Potential Gap to guarantee the safety of idealized quadrupedal models, with better robustness to nuisance factors and translation to actual robot models. Stabilizing perception space for collision checking offers additional safety to robots with internal motion artifacts. Both simulation and real experiments are conducted on our A1 quadrupedal robot to demonstrate the improved safety performance of GPF-BG. In future work, terrain traversability and multi-modal planning for different gait primitives will be integrated in this vision-based navigation framework. Safely navigating on different terrains becomes possible in the all-in-one system.

REFERENCES

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[2] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[3] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[4] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4102–4109.

[5] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[6] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, 2018.

[7] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[8] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[9] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-aware trajectory optimization and control for agile quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7755–7762, 2022.

[10] H. Zhu, D. Wang, N. Boyd, Z. Zhou, L. Ruan, A. Zhang, N. Ding, Y. Zhao, and J. Luo, "Terrain-perception-free quadrupedal spinning locomotion on versatile terrains: Modeling, analysis, and experimental validation," *Frontiers in Robotics and AI*, vol. 8, 2021.

[11] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4736–4741.

[12] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4571–4576.

[13] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[14] M. Brandao, O. B. Aladag, and I. Havoutis, "Gaitmesh: controller-aware navigation meshes for long-range legged locomotion planning in multi-layered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3596–3603, 2020.

[15] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3664–3671.

[16] P. Fernbach, S. Tonneau, A. Del Prete, and M. Taïx, "A kinodynamic steering-method for legged multi-contact locomotion," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3701–3707.

[17] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3829–3836.

[18] M. Chignoli, S. Morozov, and S. Kim, "Rapid and reliable trajectory planning involving omnidirectional jumping of quadruped robots," *arXiv preprint arXiv:2111.13648*, 2021.

[19] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2132–2139.

[20] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, *et al.*, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3306–3313.

[21] Z. Zhou, D. J. Lee, Y. Yoshinaga, S. Balakirsky, D. Guo, and Y. Zhao, "Reactive task allocation and planning for quadrupedal and wheeled robot teaming," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 2110–2117.

[22] R. Xu, S. Feng, and P. A. Vela, "Potential gap: A gap-informed reactive policy for safe hierarchical navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8325–8332, 2021.

[23] S. Feng, F. Lyu, J. Ha Hwang, and P. A. Vela, "Ego-centric stereo navigation using stixel world," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 201–13 207.

[24] J. S. Smith, S. Feng, F. Lyu, and P. A. Vela, *Real-Time Egocentric Navigation Using 3D Sensing*. Cham: Springer International Publishing, 2020, pp. 431–484.

[25] V. Dimitrov *et al.*, "Hierarchical navigation architecture and robotic arm controller for a sample return rover," in *T-SMC*, Oct 2013, pp. 4476–4481.

[26] M. Teiner, I. Rojas, K. Goser, and O. Valenzuela, "A hierarchical fuzzy steering controller for mobile robots," in *Int. Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2003, pp. 7–12.

[27] J. Guldner, V. I. Utkin, and R. Bauer, "Mobile robots in complex environments: a three-layered hierarchical path control system," in *IROS*, vol. 3, Sep. 1994, pp. 1891–1898.

[28] T. Y. Teck, M. Chitre, and P. Vadakkepat, "Hierarchical agent-based command and control system for autonomous underwater vehicles," in *ICoIAS*, June 2010, pp. 1–6.

[29] J. Warnke, A. Shamsah, Y. Li, and Y. Zhao, "Towards safe locomotion navigation in partially observable environments with uneven terrain," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 958–965.

[30] Y. Zhao, Y. Li, L. Sentis, U. Topcu, and J. Liu, "Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments," *The International Journal of Robotics Research*, vol. 41, no. 8, pp. 812–847, 2022.

[31] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *RA-M*, vol. 4, no. 1, pp. 23–33, March 1997.

[32] C. Rösmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *ECC*, July 2015, pp. 3352–3357.

[33] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm:follow the gap method," in *RAS*, vol. 60, no. 9, July 2012, pp. 1123–1134.

[34] M. Mujahad, D. Fischer, B. Mertsching, and H. Jaddu, "Closest gap based (cg) reactive obstacle avoidance navigation for highly cluttered environments," in *IROS*, 2010, pp. 1805–1812.

[35] M. Mujahed, H. Jaddu, D. Fischer, and B. Mertsching, "Tangential closest gap based (tcg) reactive obstacle avoidance navigation for cluttered environments," in *SSRR*, 2013, pp. 1–6.

[36] M. Mujahed, D. Fischer, and B. Mertsching, "Safe gap based (sg) reactive navigation for mobile robots," in *ECMR*, 2013, pp. 325–330.

[37] M. Mujahed and B. Mertsching, "The admissible gap (ag) method for reactive collision avoidance," in *ICRA*, 2017, pp. 1916–1921.

[38] J. S. Smith, R. Xu, and P. Vela, "egoTEB: Egocentric, Perception Space Navigation Using Timed-Elastic-Bands," in *ICRA*, 2020, pp. 2703–2709.

[39] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1758–1765, 2019.

[40] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[41] J. wung Choi, R. E. Curry, and G. H. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World*

Congress on Engineering and Computer Science 2008, pp. 158–166, 2008.

[42] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *Journal of Computational Science*, vol. 25, pp. 339–350, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877750317308906

[43] H. A. Satai, M. M. A. Zahra, Z. I. Rasool, R. S. Abd-Ali, and C. I. Pruncu, "Bézier curves-based optimal trajectory design for multirotor uavs with any-angle pathfinding algorithms," *Sensors*, vol. 21, no. 7, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/7/2460

[44] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1882–1891, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2020.0355

[45] V. Hassani and S. V. Lande, "Path planning for marine vehicles using bézier curves," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 305–310, 2018, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318322092

[46] B. Ingersoll, J. Ingersoll, P. DeFranco, and A. Ning, "Uav path-planning using bezier curves and a receding horizon approach," in *AIAA Modeling and Simulation Technologies Conference*, 06 2016.