Enhancing Feature Tracking Reliability for Visual Navigation using Real-Time Safety Filter

Dabin Kim*, Inkyu Jang*, Youngsoo Han, Sunwoo Hwang, and H. Jin Kim

***Abstract*— Vision sensors are extensively used for localizing a robot's pose, particularly in environments where global localization tools such as GPS or motion capture systems are unavailable. In many visual navigation systems, localization is achieved by detecting and tracking visual features or landmarks, which provide information about the sensor's relative pose. For reliable feature tracking and accurate pose estimation, it is crucial to maintain visibility of a sufficient number of features. This requirement can sometimes conflict with the robot's overall task objective. In this paper, we approach it as a constrained control problem. By leveraging the invariance properties of visibility constraints within the robot's kinematic model, we propose a real-time safety filter based on quadratic programming. This filter takes a reference velocity command as input and produces a modified velocity that minimally deviates from the reference while ensuring the information score from the currently visible features remains above a user-specified threshold. Numerical simulations demonstrate that the proposed safety filter preserves the invariance condition and ensures the visibility of more features than the required minimum. We also validated its real-world performance by integrating it into a visual simultaneous localization and mapping (SLAM) algorithm, where it maintained high estimation quality in challenging environments, outperforming a simple tracking controller.**

I. Introduction

Vision sensors are widely used for self-localization in mobile robots. Visual Odometry (VO) and Visual Simultaneous Localization and Mapping (V-SLAM) are extensively researched in both the computer vision and robotics fields. State-of-the-art visual pose estimation algorithms, such as ORB-SLAM [1] and VINS-Mono [2], have proven to be highly effective.

While most research has traditionally focused on improving the accuracy and robustness of visual estimation using available image data, recent studies have begun examining the impact of image data quality on the performance of vision-based localization algorithms. The quality of image data is often influenced by the camera's trajectory. For example, when a camera follows a trajectory that captures textureless surfaces (e.g., plain walls), the accuracy of VO may decrease due to the lack of sufficient visual features. This challenge has sparked a growing interest in perception-aware planning and control, where the camera trajectory is adjusted to ensure high-quality visual data for reliable localization. This idea has led to several studies that integrate visual estimation considerations into advanced motion planning and control strategies.

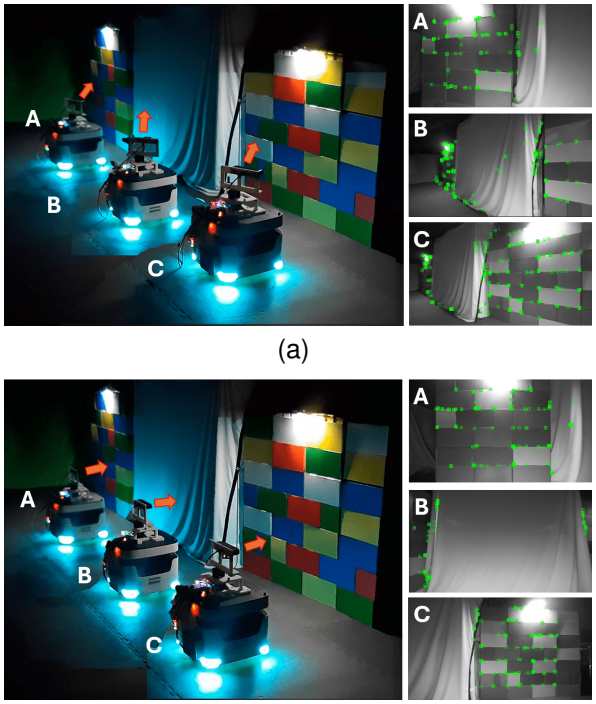

Fig. 1. The result of experiments with (a) the proposed safety filter and (b) the baseline controller. For each experiment, the robots at three timestamps (A,B,C) are visualized along with their corresponding on-board images, with features visualized in ORB-SLAM2. The detected features are represented with green dots. The proposed safety filter adaptively adjusts the control input to maintain sufficient tracking features, as demonstrated by the camera heading (arrow) and onboard image at timestamp B. In contrast, the baseline controller struggles with texture-poor surfaces, where fewer features are trackable. A detailed explanation and analysis are provided in Section V.

Belief-space planning and optimal control methods [3] [4], which rely on explicit state estimation and uncertainty modeling, are often unsuitable for real-time control in modern V-SLAM systems due to the high computational load of tasks such as bundle adjustment involving hundreds of landmarks. To address this limitation, recent research has explored incorporating information directly extracted from image inputs, such as feature points, into the planning and control loop. These algorithms typically frame the problem as a multi-objective optimization, where perception requirements become an additional objective for the controller. However, this multi-objective approach introduces significant challenges. Balancing path optimality with maintaining sufficient visual information is difficult, and comparing Pareto-optimal solu-

* The first two authors contributed equally to this work.

The authors are with the Department of Aerospace Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: dabin404@snu.ac.kr, janginkyu.larr@gmail.com, swsw0411@snu.ac.kr, cat7945@snu.ac.kr, hjinkim@snu.ac.kr, corresponding author: H. Jin Kim).

tions to find the best trade-off adds complexity. Additionally, prioritizing visual information can often degrade trajectory quality, potentially leading to numerical instability when the multiple objectives are in conflict.

In this paper, rather than adopting a multi-objective optimization approach, we address the problem from a safety-critical control perspective. Instead of prioritizing the optimization of the estimation algorithm's performance, our focus is on ensuring sufficient visual information for stable localization by framing it as a constraint in the optimization process. This approach is based on the observation that once a sufficient amount of information is obtained, adding more does not significantly enhance estimation performance.

Building on the theory of safety filters which ensure the forward invariance of a safety set, we developed a visibility maintenance condition designed to guarantee a sufficient number of visual features within an image, despite a limited field of view. From these visibility maintenance conditions, we formulate a quadratic program (QP)-based safety filter that guarantees feasibility and supports real-time computation. Unlike previous works on visibility-constrained control such as [5], which focused on maintaining fixed targets within a limited field of view, our approach accommodates newly observed or lost features dynamically. To the best of the authors' knowledge, this is the first attempt to develop a safety filter for perception-aware control in visual navigation that guarantees feasibility. We conduct numerical evaluations to validate that the designed safety filter can maintain a user-defined threshold for the minimum number of visual features. Additionally, experimental results are presented to demonstrate the efficacy of the proposed method using real-world data and a visual estimator.

II. Related Work

A. Planning and Control for Reliable Visual Navigation

The importance of acquiring rich keypoints in visual navigation has been well recognized since the early days of visual SLAM research [6]. With advancements in path planning and control, there has been a growing focus on integrating visual navigation capabilities into these algorithms.

Belief Space Planning (BSP) addresses state uncertainty by modeling it as a probability distribution, allowing objectives like collision avoidance to be treated as chance constraints. For instance, estimators with explicit uncertainty representation, such as the Extended Kalman Filter (EKF), have been combined with Rapidly-exploring Random Trees (RRT) to manage uncertainty in path planning [7]. In vision-based systems, computationally intensive techniques such as bundle adjustment are used to estimate future states. Integrating these methods into the core of planning and control algorithms is often impractical for real-time control due to their high computational cost [8] [3]. To mitigate this computational burden, metrics like the Fisher Information Matrix [9] or the observability Gramian [10] are employed as proxies for explicit uncertainty computation and are used as objective functions in optimal control algorithms.

In contrast, vision-based heuristics are designed to assess the quality of visual navigation in a computationally efficient manner, often derived directly from pixel measurements. For instance, the visibility of points of interest can be formulated as a constraint in controller design [11] [5], which is effective for tracking fixed landmarks but not suitable for dynamic visual navigation with feature tracking. Metrics such as the number of visible features [12] and co-visible features [13] are used approximations of visual estimation quality, which are integrated into trajectory optimization. Our work aligns closely with this latter approach, operating directly in image space. However, while trajectory optimization must balance objectives such as smoothness, collision avoidance, and feature visibility, we adopt a safety-critical control method that treats reliable feature tracking as a constrained control problem.

B. Safety-Critical Control with Perception Objective

The growing emphasis on autonomous navigation systems has heightened the demand for safety, particularly with formal guarantees such as collision avoidance. Safety-critical control techniques including Hamilton-Jacobi reachability and control barrier functions (CBFs) have been developed to address these needs. These techniques now increasingly integrate perception systems for broader applications.

For handling state uncertainty with estimators in the loop, measurement-robust CBFs [14] address safety constraints under measurement errors, while co-designs of observers and controllers [15] are also explored. In vision-based systems, the challenge of high-dimensional image inputs is tackled by learning-based CBFs trained on RGB-D images to avoid collisions with arbitrarily shaped objects [16], and BarrierNet [17], which uses differentiable CBFs for tasks like end-to-end driving. CBFs have also been designed for environmental representations such as point clouds from depth cameras [18] and neural radiance fields [19] for collision avoidance.

Beyond collision avoidance, safety-critical control can also be used to maintain the visibility of points of interest, which is essential in applications like visual servoing [20] [21] and teleoperation [5], ensuring that targets remain within the camera's field of view. As a new application for a safety-critical control, we develop a safety filter tailored for reliable visual navigation, with safety requirements based designed to ensure a sufficient number of visible image features.

III. Real-Time Constraint Satisfaction using QP-Based Safety Filter

Assume a continuous-time time invariant nonlinear system model

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in U \subseteq \mathbb{R}^m$ is the input. The set U is assumed to be a convex polytope in the $\mathbb{R}^m$ space, i.e., there exist a matrix A_u and a vector b_u with appropriate sizes such that $U = \{u \in \mathbb{R}^m : A_u u \leq b_u\}$. We assume that for every reachable state x in the state space, there exists an input $u \in U$ (possibly as a function of x) such that $f(x) + g(x)u = 0$. This u is called the *stopping input*

that brings the system to an instantaneous complete stop. Relaxing this sudden stop assumption is an interesting future research direction which will be discussed in the conclusion section.

Given a set of allowed states $C \subseteq \mathbb{R}^n$, we want to generate a feedback law $u(t,x) \in U$ as a function of time t and state x, such that the system permanently resides in C, i.e., C is forward invariant. Suppose there exist a finite number of continuously differentiable functions $h_i : \mathbb{R}^n \to \mathbb{R}$, $\forall i \in I$, where I is the index set, such that $C = \{x \in \mathbb{R}^n : h_i(x) \geq 0\}$, and $\partial_x h_i(x) \neq 0$ if $h_i(x) = 0$. Then, Nagumo's theorem [22, Section 4.2] tells that if u is chosen such that

$$h_i(x) = 0 \to \partial_x h_i(x) \cdot (f(x) + g(x)u) \geq 0, \ \forall i \in I, \quad (2)$$

then the set C will be rendered forward invariant.

In order to build a feedback control strategy that best tracks the given reference input while satisfying (2) and avoiding discontinuity, one can let $u(t,x)$ be the optimal solution to the following quadratic program:

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \ & (u - u_{\text{ref}}(t,x))^\top R (u - u_{\text{ref}}(t,x)) \\ \text{s.t. } & \dot{h}_i(x,u) \geq -\alpha_i h_i(x), \ \forall i \in I \\ & A_u u \leq b_u, \end{aligned} \quad (3)$$

where $R \in \mathbb{R}^{m \times m}$ is a symmetric positive definite weight matrix, $\dot{h}_i(x,u) = \partial_x h_i(x) \cdot (f(x) + g(x)u)$ is the time derivative of $h_i(x)$ given state x and input u, α_i-s are positive reals. Note that given a strictly positive definite R, (3) admits a unique solution which satisfies (2). Additionally, the optimization is feasible given $x \in C$, since the stopping input is one feasible solution to the optimization. Thus, (3) can be called a *safety filter* [23], [24] in the sense that it selects the input from the set of safe inputs $U_f = \{u : u \text{ is a feasible solution to (3)}\}$ that minimally deviates from the reference. This approach can be regarded as a special case of CBF-based quadratic programs (CBF-QP) [25] where the instantaneous brakability assumption allows to employ multiple CBFs in a set-intersection manner without suffering from *leaking corner* issues [26].

IV. Visibility Maintenance using Safety Filter

A. Differential Kinematics

Let the robot's configuration space be $Q \subseteq \mathbb{R}^n$. We model the robot's differential kinematics as follows:

$$\dot{q} = J(q)v, \quad (4)$$

where $J(q) : Q \to \mathbb{R}^{n \times m}$ is the Jacobian matrix which we assume continuous with respect to q, $v \in V \subseteq \mathbb{R}^m$ is the input, and V is the input constraint set which is assumed to be a convex polytope such that $V \ni 0$ (i.e., it satisfies the instantaneous brakability condition).

We consider a vision sensor attached rigidly to the robot frame (so that its pose $T \in SE(3)$ is given as a continuously differentiable function of the robot's configuration q), which is capable of detecting positions of point landmarks within its region of detection. To elaborate, let p be the position of a landmark seen in the vision sensor frame. The condition

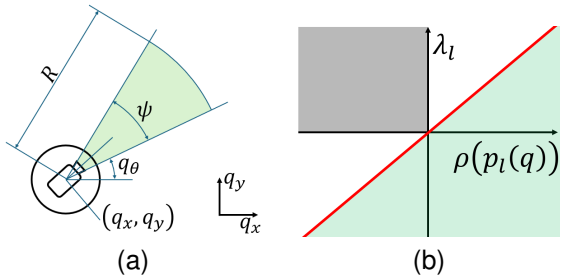

Fig. 2. (a) The robot configuration and the onboard camera's field of view for the running example. The camera is mounted on the two dimensional ground robot, captures any landmark within its field of view, represented by the light green region. The field of view is defined by the angle of view, ψ, and the sensing range, R. (b) A graphical illustration of equivalence between (14) and (16). The gray region shows where (14) is violated. For a fixed $\mu \in [0,1]$, the feasible region of (16) forms a half-space with the origin on its boundary. The union of all possible half-spaces aligns with the feasible set of (14), represented by the white region.

for the landmark to be seen from the sensor can be written as $\rho(p) \geq 0$, where $\rho : \mathbb{R}^3 \to \mathbb{R}^d$ is the continuous and differentiable constraint function.

The motion of the sensor frame can be written using its body linear and angular velocities (i.e., twist). If we assume the landmarks are fixed to the world frame (i.e., stationary), the landmark position with respect to the sensor p is controllable by the body twist and follows the dynamics

$$\dot{p} = -\omega_s \times p - v_s, \quad (5)$$

where $\omega_s \in \mathbb{R}^3$ and $v_s \in \mathbb{R}^3$ are the sensor's body angular and linear velocities, and $\times$ is the vector cross product in 3D. Since T is a continuously differentiable function of q, we can then write ω_s and v_s as a linear function of the control input v as follows:

$$\omega_s = J_\omega(q)v, \ \ v_s = J_v(q)v, \quad (6)$$

where J_ω and J_v are matrix-valued continuous functions of q with appropriate sizes.

Running Example (Ground robot)**.** In this section, as a running example, we consider the following ground robot (Fig. 2) with a camera attached on it:

$$\dot{q} = \frac{d}{dt}\begin{bmatrix} q_x \\ q_y \\ q_\theta \end{bmatrix} = v = \begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix}, \quad (7)$$

where q_x, q_y are the horizontal and vertical positions of the robot base, q_θ is the orientation (rotation angle with respect to the x axis) of the camera, v_x, v_y, v_θ are control inputs. It is straightforward to find that the landmark point kinematics (5) can be written as

$$\dot{p} = \frac{d}{dt}\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} -\cos q_\theta & -\sin q_\theta & p_y \\ \sin q_\theta & -\cos q_\theta & -p_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix}. \quad (8)$$

Here, p_x and p_y are the horizontal and vertical positions of the landmark seen from the camera frame, respectively.

B. Visibility Maintenance

Let L be the set of N landmarks in three-dimensional space, where the position of each landmark $l \in L$ relative to the vision sensor is $p_l(q)$. A landmark l is visible if $\rho(p_l(q)) \geq 0$, and the set of visible landmarks is denoted as $L(q) := \{l \in L : \rho(p_l(q)) \geq 0\}$.

Running Example (Ground robot, continued)**.** We assume that the onboard camera can detect landmarks within the field of view as shown in Fig. 2 (a). Given the field of view, the ρ function can be defined as follows:

$$\rho(p) = \begin{bmatrix} [\sin\psi/2, \cos\psi/2]^\top p \\ [\sin\psi/2, -\cos\psi/2]^\top p \\ R - \|p\|_2 \end{bmatrix} \tag{9}$$

where ψ is the angle of view, R is the sensing range. In this example, we let $R = 1$, $\psi = 1$ rad.

Given all above, we want the robot's motion to satisfy the following constraint:

$$w(q) := \sum_{l \in L(q)} w_l \geq W, \tag{10}$$

where $w_{(\cdot)} \geq 0$ are the weights, $w(q)$ represents the overall landmark tracking quality score, w_l denotes the contribution of landmark l to this quality, and W is the minimum score the robot is required to achieve throughout the mission. For example, if the goal is to maintain visibility to at least M landmarks, we can let $w_l = 1$ for all $l \in L$ (so that $w(q)$ counts the number of currently visible landmarks), and W any positive number between $M - 1$ and M.

Unfortunately, since $w(q)$ is not everywhere differentiable, it is impossible to construct a QP-based safety filter like (3). Instead, as a way around, we introduce an auxiliary state variable $\lambda = [\lambda_1, \cdots, \lambda_N]^\top \in \mathbb{R}^N$ and a *smoothened* version of score constraint

$$W \leq \hat{w}(q, \lambda) = \sum_{l \in L} \lambda_l w_l \leq w(q). \tag{11}$$

The first inequality $W \leq \hat{w}$ is continuously differentiable with respect to λ and has no dependency on q. For the second inequality $\hat{w} \leq w$ to hold, we require

$$\lambda_l \leq \mathbf{1}_{l \in L(q)}(q) \tag{12}$$

where $\mathbf{1}_{\phi(\xi)}(\xi)$ is the indicator function that returns 1 if the statement $\phi(\xi)$ is true and 0 otherwise. This can be rewritten as the intersection of two constraints for every $l \in L$:

$$\lambda_l \leq 1, \tag{13}$$

$$\lambda_l > 0 \;\rightarrow\; \rho(p_l(q)) \geq 0. \tag{14}$$

The first inequality (13) is continuously differentiable, and Eq. (14) is equivalent to

$$\lambda_l \leq 0 \;\vee\; \rho(p_l(q)) \geq 0, \tag{15}$$

which holds if and only if

$$\exists \mu_l \in [0, 1], \quad -\mu_l \lambda_l + (1 - \mu_l)\rho(p_l(q)) \geq 0. \tag{16}$$

The proof for this claim is straightforward and hence omitted in this paper. A graphical explanation to this can be found in Fig. 2 (b). To combine everything into a continuously differentiable setting, we let $\mu = [\mu_1, \cdots, \mu_N]^\top \in \mathbb{R}^N$ be another auxiliary state variable of the system. The auxiliary state λ and μ are *controlled* through the dynamics

$$\dot{\lambda} = v_\lambda, \quad \dot{\mu} = v_\mu, \tag{17}$$

where $v_\lambda, v_\mu \in \mathbb{R}^N$ are virtual inputs that are numerically integrated to obtain the actual λ and μ values.

C. QP-based Safety Filter

In summary, we have the following set of state constraints

$$\begin{aligned} h_1(x) &= \sum_{l \in L} \lambda_l w_l - W \geq 0, \\ h_{2,l}(x) &= 1 - \lambda_l \geq 0, \ \forall l \in L, \\ h_{3,l}(x) &= -\mu_l \lambda_l + (1 - \mu_l)\rho(p_l(q)) \geq 0, \ \forall l \in L, \\ h_{4,l}(x) &= \mu_l \geq 0, \ \forall l \in L, \\ h_{5,l}(x) &= 1 - \mu_l \geq 0, \ \forall l \in L, \\ h_6(x) &= c(q), \end{aligned} \tag{18}$$

with $x = (q, \lambda, \mu) \in \mathbb{R}^{n+2N}$ being the augmented state. The constraints h_1 through $h_{5,l}$ are derived from Section IV-B, and h_6 is the collision avoidance constraint encoded as a function of robot state $c(q)$ which we assume continuously differentiable. For example, one can model the robot as a sphere and let $c(q) = s(q) - r$, where $s(q)$ is the (signed) distance from the robot to the nearest obstacle, $r > 0$ is the radius of the robot.

These state constraints are continuously differentiable with respect to the augmented state x and a safety filter in the form (3) can be implemented:

$$\begin{aligned} \min_{u \in U} \quad & (u - u_{\text{ref}}(t, x))^\top R (u - u_{\text{ref}}(t, x)) \\ \text{s.t.} \quad & \dot{h}_{(\cdot)}(x, u) \geq -\alpha_{(\cdot)} \cdot h_{(\cdot)}(x) \end{aligned} \tag{19}$$

where the inequality constraint should be satisfied for all $h_{(\cdot)}$ functions in (18). Here, we let $u = (v, v_\lambda, v_\mu) \in U = V \times \mathbb{R}^N \times \mathbb{R}^N \subseteq \mathbb{R}^{m+2N}$ be the augmented input and $u_{\text{ref}}(t, x) = (v_{\text{ref}}(t, q), 0, 0)$, $R = \text{blkdiag}(R_q, k_\lambda 1_N, k_\mu 1_N)$ where v_{ref} is the reference input given from the higher-level decision maker (e.g., manual control from a human operator or a motion planning algorithm), $R_q \in \mathbb{R}^{m \times m}$ is the symmetric positive definite input cost matrix, k_λ and k_μ are positive (usually very small) weights to ensure the safety filter has a unique solution, 1_N is the identity matrix of size $N \times N$.

The time derivatives of $h_{(\cdot)}$ can be expressed as a linear function of u, thus the safety filter (19) is a QP. The λ and μ values should be calculated by integrating (17). In case $p_l(q)$ and its derivative are not straightforward to directly compute, one can evaluate it in an indirect manner by forward integrating to get the following initial value ODE, which follows directly from (5) and (6):

$$\begin{aligned} \dot{p}(t) &= -J_\omega(q(t))v(t) \times p(t) - J_v(q(t))v(t), \\ p(t_0) &= p_l(q(t_0)) \end{aligned} \tag{20}$$

where $p_l(q(t_0))$ is the position of the landmark in sensor frames when initially observed at time t_0, and p is a shorthand for $p_l \circ q$. It can be seen that if $h_{(\cdot)}(x) \geq 0$, the optimization (19) is feasible since $u = 0$ makes $\dot{h}_{(\cdot)} = 0$ and thus is one feasible solution.

Running Example. (Ground robot, continued) In the running example, the mission of the robot is to track the reference input, while maintaining visibility to at least 5 landmarks. Thus, we let $W = 4.5$, $w_l = 1$. We place $N = 30$ landmarks at random positions near the origin. The reference input is given as

$$v_{\text{ref}}(t, q) = \begin{bmatrix} -\sin t + 2(\cos t - q_x) \\ \cos t + 2(\sin t - q_y) \\ 0 \end{bmatrix} \tag{21}$$

which makes the robot to track a circular trajectory around the origin at speed 1. In this example, we ignore collision avoidance constraints (h_6 in (18)). Other parameter values are set as follows: $\alpha_{(\cdot)} = 1$, $V = [-2, 2] \times [-2, 2] \times [-1, 1]$, $R = \text{diag}(1, 1, 0.001)$, $k_\lambda = k_\mu = 0.001$. The third component of R corresponding to v_θ is set to a small number to allow the camera to rotate accordingly to keep the landmarks in sight.

D. Initialization and Handling Limited Observability

To take advantage of the nonnegativity-preserving property of the safety filter, it is very important to initialize the safety filter with a valid augmented state x so that $h_{(\cdot)} \geq 0$. Suppose the robot starts at an initial condition $q(t_0)$ such that $w(q(t_0)) \geq W$ and $c(q(t_0)) \geq 0$. This means that the robot has view to a sufficient amount of information from the landmarks and is at a collision-free position. Then, one can easily find that the initialization $\lambda_l(t_0) = \mathbf{1}_{l \in L(q(t_0))}(l)$, $\mu_l(t_0) = \mathbf{1}_{l \notin L(q(t_0))}(l)$ is a valid choice that $h_{(\cdot)}(x) \geq 0$.

In real deployment, the landmark positions p_l are typically not *a priori* known and the robot can observe only the currently visible landmark positions at discrete times. Suppose that the observations occur at times $t_0 < t_1 < \cdots$. Whenever $t = t_i$ for an integer $i \geq 0$, the robot newly observes p_l such that $l \in L_i = L(q(t_i))$. During the interval $t \in [t_i, t_{i+1})$ for every $i \in \{0, 1, \cdots\}$, we require the robot to run the safety filter with L_i instead of L, with re-initialization $\lambda_l(t_i) = \mathbf{1}_{l \in L_i}(l) = 1$, $\mu_l(t_0) = \mathbf{1}_{l \notin L_i}(l) = 0$. This will introduce a sudden jump in the $h_{(\cdot)}$ values at time $t = t_i$, however, here we show that the jumps preserve nonnegativity of $h_{(\cdot)}$ values. Let $x_i^- = (q_i, \lambda_i^-, \mu_i^-)$ and $x_i^+ = (q_i, \lambda_i^+, \mu_i^+)$ be the augmented states before and after the jump at time t_i ($i \in \{1, 2, \cdots\}$), respectively. Note that the robot configuration q does not jump. Suppose $h_{(\cdot)}(x_i^-) \geq 0$. Firstly, h_1 value always jump *upwards*, i.e., $h_1(x_i^+) \geq h_1(x_i^-) \geq 0$, since $h_1(x_i^-) \leq w(q_i) - W$ (the nonnegativity of $h_{(\cdot)}(x_i^-)$ ensures this), and $h_1(x_i^+) = \sum_{l \in L_i} w_l - W = w(q_i) - W$. Next, it is straightforward to find the values of $h_{2,l}$ through $h_{5,l}$ are nonnegative after the update. Finally, h_6 only depends on q which does not jump, therefore $h_6(x_i^+) = h_6(x_i^-) \geq 0$.

In summary, if the initial conditions are such that $w(q(t_0)) \geq W$ and $c(q(t_0)) \geq 0$, then the robot will start from a feasible condition that satisfies (18). Moreover, for every $i \in \{0, 1, \cdots\}$, the safety filter will ensure the constraint satisfaction throughout $t \in [t_i, t_{i+1})$, and the jump at $t = t_{i+1}$ will set the $h_{(\cdot)}$ value to a nonnegative one, after which the safety filter can resume from a feasible initial condition.

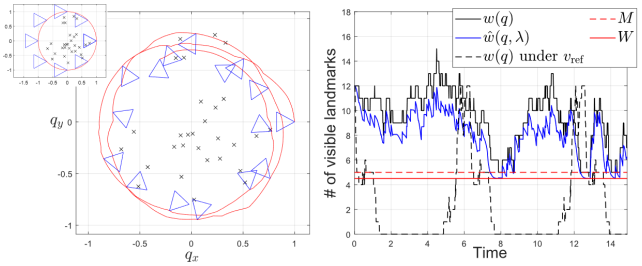

Fig. 3. Simulation result for the running example. (left) The resulting trajectory of the robot. The position trajectory (q_x, q_y) is depicted as red curve, the camera poses are drawn using blue triangles. It can be seen that the robot takes a path that differs from the reference (shown in the left top corner) to maintain visibility to at least 5 landmarks. (right) The time history of $w(q)$, $\hat{w}(q, \lambda)$ values. The relation $W \leq \hat{w}(q, \lambda) \leq w(q)$ holds throughout the simulation.

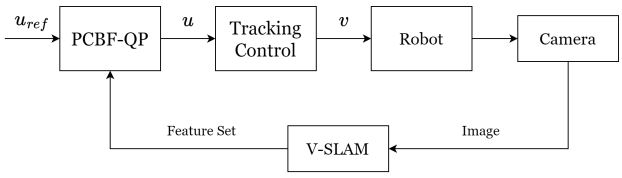

Fig. 4. The diagram illustrated for the control structure for vision-based robot with the proposed safety filter.

Running Example (Ground robot, simulation result)**.** With the abovementioned setting and initialization, we simulated the safety filter using MATLAB. The results are depicted in Fig. 3. In the results, it can be clearly seen that the $\hat{w}$ value always lower bounds the number of visible landmarks w and lower bounded by W, resulting in $w(q) \geq 5$ throughout the simulation. The robot takes the path that provides visibility to at least 5 landmarks, while minimally deviating from the reference input given. Note that the re-initialization introduces sudden jumps in the $\hat{w}(q, \lambda)$ values, but only in a way that the state constraints are satisfied.

V. Evaluation

The proposed safety filter is validated on real hardware in an experimental setup in a vision-challenging environment.

A. Mission Objective and Hardware Configuration

The mission is to inspect a wall using a stereo camera, which includes both texture-rich and texture-poor regions, as shown in Fig. 5 (a). The objective is to move along the wall while keeping the camera oriented perpendicular to the surface, ensuring an optimal view of the inspection area. However, the texture-poor region poses a challenge for visual navigation.

The hardware configuration is depicted in Fig. 5 (b). The base robot is the Former robot of ROAS Inc., operates using

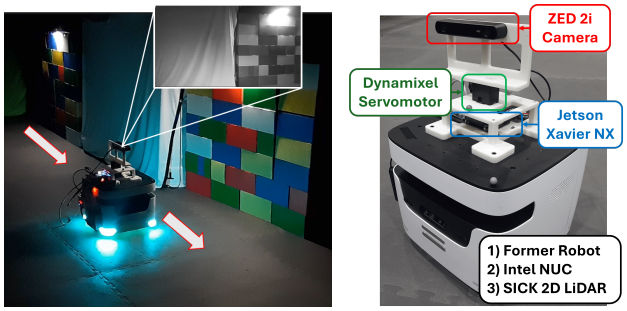

Fig. 5. (a) The experiment scenario where the robot inspects the wall moving through the texture-poor region and (b) the hardware configuration used for the hardware experiment.

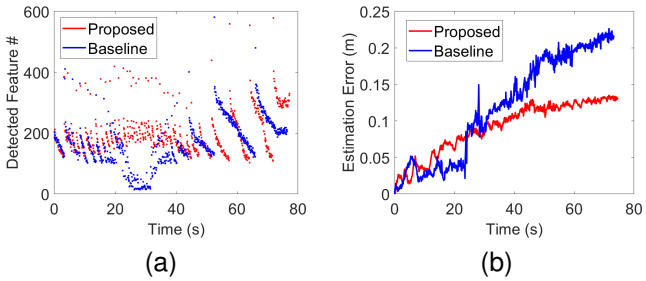

Fig. 6. Comparison of the result of the visual SLAM algorithm in between the proposed safety filter (red) and the baseline controller (blue). (a) The number of tracked features and (b) the state estimation error acquired from the ORB-SLAM2 algorithm.

a differential wheeled drive system. A ZED2i stereo camera is mounted on a rig equipped with a servomotor, enabling rotation in the same axis with the rotation of the base robot. An NVIDIA Jetson Xavier NX computer is paired with the stereo camera for real-time processing. Additionally, a SICK 2D LiDAR is mounted on the base robot for obstacle detection. All system components are connected via Ethernet communication.

B. Implementation Details

The control structure with the proposed safety filter and hardware is shown in Fig. 4. The robot state and the velocity are defined as $q = [q_x, q_y, \theta_r, \theta_m]^T$ and $v = [v_r, \omega_r, \omega_m]^T$, where θ_r, v_r and ω_r are heading angle, velocity and angular velocity of the base robot and θ_m and ω_m are the heading angle and angular velocity of the servomotor. The Jacobian matrix J is $\begin{bmatrix} \cos\theta_r & 0 & 0 \\ \sin\theta_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, and the heading angle of the camera is $\theta_c = \theta_r + \theta_m$.

The reference linear velocity and angular velocity (v_r, ω_r) for the base robot, and the reference angular velocity $(\omega_{s,r})$ for the servomotor are given to the safety filter. v_r is set as positive constant value and ω_r to zero when the heading θ_r is set to the direction parallel to the wall. The servomotor is controlled by a PID controller to regulate the camera to be aligned with the wall's normal vector.

For visual navigation, we employ ORB-SLAM2 [1] with stereo images. Each extracted feature is measured as $m = [m_u, m_v, m_d]^T = [f_x \frac{p_x}{p_z} + c_x, f_y \frac{p_y}{p_z} + c_y, d]^T$, where (f_x, f_y) represents the focal lengths, (c_x, c_y) denotes the principal point, and d is the depth measurement. The visibility condition ρ is defined as $m_u \in [0, I_w], m_v \in [0, I_h]$, and $m_d \in [r_{min}, r_{max}]$ where I_w and I_h are the width and height of the image, and r_{min} and r_{max} are the depth thresholds for valid detection. Since the number of inequality constraints linearly increases to the number of features, we limit the features passed to the safety filter to $N_{max} = 50$ to meet the real-time requirement for solving the QP problem (19) by sampling N_{max} number of features from the total feature set. The signed distance function from h_6 in (18) is obtained from 2D LiDAR data.

C. Results

In the baseline algorithm, the reference velocity is applied directly to the robot's tracking controller, with the servomotor keeping the camera perpendicular to the wall. As shown in the onboard images of Fig. 1 (a), the robot moves through a texture-poor region, causing a shart drop in visible features and an increase in estimation error (Fig. 6).

In contrast, the proposed safety filter adjusts the control input to maintain a sufficient number of visible features, even when passing through feature-poor regions. It rotates the camera toward feature-rich regions to ensure continuous tracking, returning to the orientation close to the desired orientation when the robot reaches a feature-rich area. As a result, a sufficient number of features are kept tracked as in Fig. 6 (a). We observe that, in this scenario, our algorithm successfully maintains a sufficient number of visible features, implying it can be used along with the visual front-end with real-world data.

VI. Conclusion

Despite the popularity of vision sensors for mobile robot navigation and their successful implemention across various platforms, they remain vulnerable to issues such as low-texture environments or poor lighting conditions. To address these challenges, researchers have attempted to design trajectory planners or controllers that maximize visual information while simultaneously optimizing control objectives. However, these approaches often suffer from sensitivity to parameter choices and poor numerical stability.

Instead of using a multi-objective formulation, we proposed a safety-critical control approach to tackle this issue using a safety filter. We designed a safety condition to ensure the presence of a sufficient number of visual features in the image. The QP-based formulation of the safety filter allows for real-time implementation on onboard computers. Our experimental results demonstrate that the safety filter for reliable feature tracking can easily integrate with conventional visual odometry or SLAM algorithms, reducing the risk of catastrophic failures in visual estimation.

For future research, we plan to extend the formulation by incorporating inertial sensors, allowing the filter to be used with visual-inertial navigation systems.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[3] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 849–882, 2015.

[4] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[5] D. Kim, M. Pezzutto, L. Schenato, and H. J. Kim, "Visibility-constrained control of multirotor via reference governor," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 5714–5721.

[6] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[7] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 723–730.

[8] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion-and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

[9] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5986–5992.

[10] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, 2017.

[11] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[12] D. Kim, G. C. Kim, Y. Jang, and H. J. Kim, "Topology-guided path planning for reliable visual navigation of mavs," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3117–3124.

[13] X. Chen, Y. Zhang, B. Zhou, and S. Shen, "Apace: Agile and perception-aware trajectory generation for quadrotor flights," *2024 ICRA*, 2024.

[14] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6286–6291.

[15] D. R. Agrawal and D. Panagou, "Safe and robust observer-controller synthesis using control barrier functions," *IEEE Control Systems Letters*, vol. 7, pp. 127–132, 2022.

[16] H. Abdi, G. Raja, and R. Ghabcheloo, "Safe control using vision-based control barrier function (v-cbf)," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 782–788.

[17] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.

[18] M. De Sa, P. Kotaru, and K. Sreenath, "Point cloud-based control barrier function regression for safe and efficient vision-based control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 366–372.

[19] M. Tong, C. Dawson, and C. Fan, "Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 511–10 517.

[20] D. Zheng, H. Wang, J. Wang, X. Zhang, and W. Chen, "Toward visibility guaranteed visual servoing control of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1087–1095, 2019.

[21] S. Wei, B. Dai, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "Diffocclusion: Differentiable optimization based control barrier functions for occlusion-free visual servoing," *IEEE Robotics and Automation Letters*, 2024.

[22] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[23] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[24] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2023.

[25] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[26] D. Lee, M. Chen, and C. J. Tomlin, "Removing leaking corners to reduce dimensionality in hamilton-jacobi reachability," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9320–9326.