

MirageMarker: A Black-Box Pose Estimation Attack for Autonomous UAV Landing

Junchi Lu, Fayzah Alshammari, Shaoyuan Xie, Xiaoqing Liang, and Qi Alfred Chen

Abstract—ArUco markers are widely used in unmanned aerial vehicles (UAVs) landing for their computational efficiency and reliability. However, existing robustness studies have mainly focused on the detection under poor lighting or occlusion, while overlooking the accuracy of subsequent pose estimation. We reveal that even small corner shifts can severely degrade pose estimation accuracy. To exploit this vulnerability, we propose MirageMarker, which introduces imperceptible, adversarial corner perturbations to manipulate the estimated pose. Since classic ArUco pose estimation relies on discrete corner detection and non-differentiable Perspective-n-Point (PnP) geometric solvers, we formulate the attack using a black-box optimization strategy with multi-objective constraints to generate adversarial markers. Furthermore, we validate MirageMarker in safety-critical UAV landing scenarios, conducting comprehensive experiments in NVIDIA Isaac Sim across two representative marker board layouts.

I. INTRODUCTION

Vision-based navigation systems are a key component of modern unmanned aerial vehicles (UAVs). Compared with GPS or inertial sensors, camera-based approaches provide higher spatial accuracy and richer environmental information. This is particularly critical during the final stage of autonomous landing, where GPS accuracy is insufficient and the camera provides reliable relative pose for real-time control, as shown in Fig. 1. Fiducial markers are visual reference patterns used for identification and pose estimation. ArUco markers are a widely adopted family of fiducial markers, offering robust 6-DoF pose estimation with low computational cost and widely described as “robust, fast and simple” in practice [1], [2]. They are extensively used for autonomous landing, including in real-world systems such as Google Wing’s delivery drones [3], [4] and Meituan’s urban logistics platforms [5], [6].

Most existing studies on ArUco markers emphasize detection robustness under challenging conditions such as poor lighting, occlusion, or motion blur [7], [8]. These efforts primarily address whether a marker can be detected, but none of them consider the vulnerability that small corner offsets can propagate into significant errors in downstream pose estimation. For UAV landing, this vulnerability is of particular concern, since errors in pose estimation can directly compromise landing accuracy. Adversarial attacks on UAVs have mainly targeted deep learning-based perception modules such as tracking or navigation [9], [10], [11]. These methods rely on deep learning model differentiability,

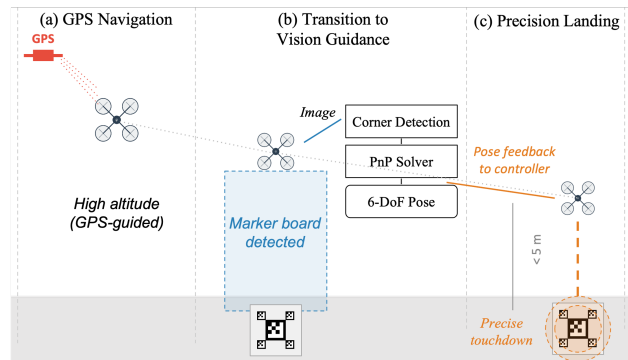


Fig. 1. UAV autonomous landing workflow illustrating the transition from GPS navigation to vision-based precision guidance using ArUco markers. The UAV switches from GPS to vision-guidance upon detecting the marker pad at lower altitudes, performing corner detection and pose estimation to achieve a precise touchdown.

whereas the ArUco pose estimation based landing pipeline is built on corner detection and Perspective- n -Point (PnP) [12] geometry, making it non-differentiable, and thus the attack can not be directly applied to such techniques.

Other studies consider cyber attacks on UAV communication and sensing channels [13], [14], [15]. For example, [15] presents stealthy false-data-injection attacks that compromise both camera images and sensor measurements used for control. However, such attacks manipulate sensing data rather than exploiting vulnerabilities specific to the ArUco pose estimation pipeline. By contrast, our perturbations act on the physical marker rather than the sensing channels, directly targeting the ArUco pose estimation pipeline while producing consistent visual measurements from the first frame. They do not generate the abrupt cross-sensor residuals that many fusion-based detectors rely on. Consequently, perturbing the geometry of ArUco markers offers a practical way to mislead pose estimation and reveals a broader security problem concerning the integrity of fiducial marker-based pose estimation pipeline under adversarial manipulation of marker geometry. To the best of our knowledge, we are the first to systematically investigate this problem space, focusing on ArUco corner perturbations in this work. An important question is whether the resulting pose estimation errors can lead to failures in UAV landing, which requires highly accurate localization.

To answer this question, we propose MirageMarker, an adversarial framework designed to manipulate ArUco-based pose estimation. By introducing fundamental perturbations through minimal corner shifts, MirageMarker consistently induces pose estimation errors, thereby compromising UAV landing performance and leading to an unsteady landing

process and landing location errors. In our end-to-end UAV landing evaluations, MirageMarker causes landing errors of up to 12.37 m, with all attack trials exceeding a 5 m landing error. Our approach demonstrates a fundamental vulnerability that generalizes across various marker board layouts, revealing a significant end-to-end impact on autonomous landing performance within high-fidelity simulation environments. Video demonstrations and code can be found on our project website at <https://sites.google.com/view/secure-safe-ai/miragemarker>. Our contributions are:

- We present the first systematic study of vulnerabilities in fiducial marker-based pose estimation under adversarial manipulation of marker geometry, focusing on the ArUco marker-based family and corner perturbation as the concrete threat model for both representativeness and practicality in the autonomous UAV landing problem space.
- We formulate a novel black-box multi-objective attack against the non-differentiable ArUco pose estimation pipeline, inducing large pose errors while preserving detectability, physical feasibility, and visual stealth.
- We conduct systematic end-to-end landing evaluations across representative board layouts in a high-fidelity simulator, showing large touchdown deviations and that the attacks remain effective under conventional filters.

II. BACKGROUND AND THREAT MODEL

This section introduces the key components of vision-based landing and the ArUco-based pose estimation, discusses prior efforts to improve marker detection robustness, and concludes with the threat model adopted in this work.

A. Vision-Based UAV Landing and ArUco Pipeline

A typical UAV landing system must overcome the limited accuracy of GPS, which is often degraded or unavailable in GPS-denied environments, such as urban canyons, indoor facilities [16], [17]. To ensure precise and safe landing, vision-based approaches using a monocular camera and fiducial markers (e.g., ArUco, AprilTag) have become a practical solution [18], [19], [20]. In these systems, the relative pose of the UAV to the landing pad is estimated from the visual module, optionally fused with data from an inertial measurement unit (IMU) through filtering methods for robustness. As shown in Fig. 1, the landing process typically proceeds in stages, transitioning from GPS-based navigation to local vision guidance once the pad becomes visible, and finally achieving marker-based precision landing [4], [21], [22].

UAV precision landing often relies on the ArUco-based pose pipeline [4], [23], [24]. The detector localizes the marker and refines the 2D corner coordinates. A PnP solver then estimates the marker-relative 6-DoF pose using the known marker geometry. In practice, the pipeline is also sensitive to environmental factors: under low illumination, motion blur, or partial occlusion, corner localization may fail and degrade pose accuracy. To address these issues,

prior work has explored methods such as adaptive thresholding [25], image pre-processing, and deep learning based ArUco detection [7], [8], which improve robustness under challenging conditions.

B. Threat Model

Attack Goals. The goal is to compromise UAV precision landing by inducing significant pose estimation errors through minimal shifts of ArUco marker corners. Such errors translate into landing position deviations and an unstable landing process, which are safety-critical for vertical take-off and landing (VTOL) operations.

Attacker Knowledge. We assume the attacker knows the design and spatial layout of the marker board, which can often be observed at deployed landing sites. Camera models and parameters of UAVs can be obtained from manufacturer documentation or by purchasing the same platform [26], [27]. Although the attack is constructed using known camera parameters, its foundation in corner-based geometric estimation suggests that the crafted markers may also generalize to UAVs with different camera configurations.

Attack Scenario. The attack targets the final landing phase, typically when the UAV descends to an altitude of less than 5 meters above the ground [4], [21], [22]. At this stage, the navigation system switches from GPS or inertial guidance to vision-based localization, relying heavily on ArUco-based pose estimation. We assume the attacker can replace or overlay the landing pad with the crafted marker board, but has no capability to modify or inject data into the UAV's onboard sensors. This is realistic because landing pads are often physically accessible [4], [6].

Constraints. The attack must preserve successful ArUco detection so that the system continues to perform pose estimates. In addition, the marker must remain visually plausible to human observers, ensuring that the attack is stealthy in realistic deployments.

III. METHODOLOGY

The overall workflow of our attack is illustrated in Fig. 2. It consists of two main stages: (a) initial adversarial marker generation, where we derive a view-specific initialization by geometric modeling under a chosen nominal observation direction; and (b) optimization, where the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [28] refines the marker over a specified viewpoint range under a joint objective. We optimize a corner-perturbation marker to bias the ArUco PnP pose estimate toward the desired target, while regularizing the solution to keep perturbations small and smooth, maintain reliable detection, and preserve visual similarity for physical deployment.

A. Initial Adversarial Marker Generation

We first model how small corner shifts can induce significant pose estimation errors. Based on this, we propose a geometric projection attack to obtain an initial adversarial marker effective at a specific viewing angle.

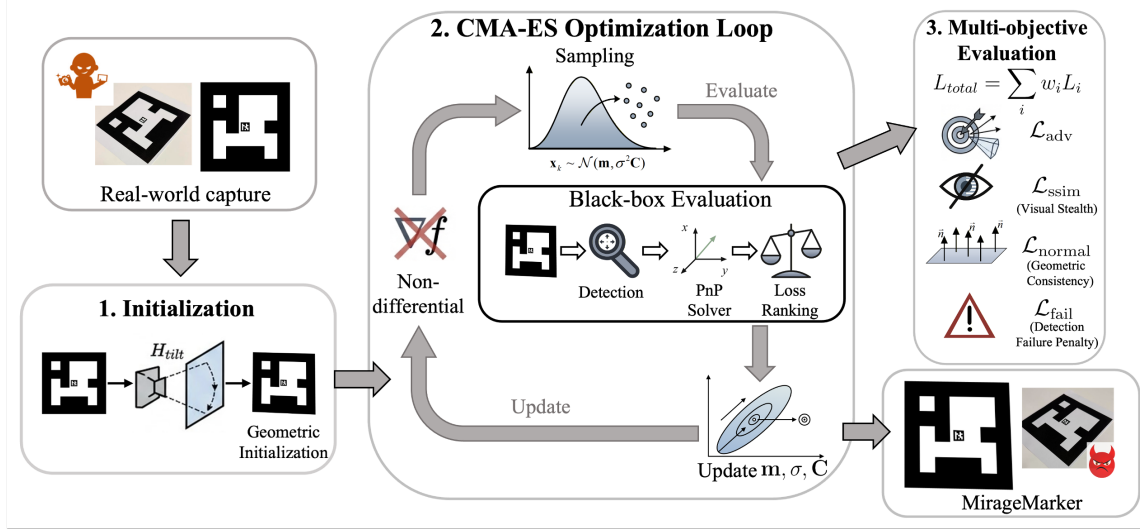


Fig. 2. Overview of MirageMarker. The attacker first records the deployed marker-board layout and recreates a clean board in a rendering space. A homography-based geometric initialization provides a seed corner perturbation for a chosen nominal view. CMA-ES then searches over corner shifts under a multi-objective score that combines the attack objective with feasibility and stealth terms, using the non-differentiable ArUco detection–PnP pipeline as the evaluator. The resulting board is printed and used to replace the original pad, enabling a practical and stealthy attack on UAV precision landing.

a) *Sensitivity of pose estimation to corner shifts:* Given the intrinsic matrix K , the 3D marker corners $\{X_i\}_{i=1}^N \subset \mathbb{R}^3$ in the marker coordinate frame, and their projections $u_i = \pi(K(RX_i + t)) \in \mathbb{R}^2$ under pose (R, t) , where $\pi([x, y, z]^T) = [x/z, y/z]^T$ is the perspective projection, pose is estimated by minimizing the reprojection error $\sum_{i=1}^N \|\hat{u}_i - u_i\|_2^2$, where $\hat{u}_i$ are the detected image corners. Let the stacked residual be $r = [\hat{u}_1 - u_1; \dots; \hat{u}_N - u_N] \in \mathbb{R}^{2N}$ and parameterize an infinitesimal pose update by a 6D twist $\delta\xi = [\delta\theta^T \ \delta t^T]^T \in \mathbb{R}^6$. A first-order linearization around (R, t) yields the Gauss–Newton system

$$J^T J \delta\xi = J^T \delta u, \quad \delta u \triangleq [\Delta u_1; \dots; \Delta u_N], \quad (1)$$

where $J = \partial r / \partial \xi \in \mathbb{R}^{2N \times 6}$ is the PnP Jacobian with respect to the pose parameters, and $\Delta u_i = \hat{u}_i - u_i$ are the corner shifts. Solving gives the first-order sensitivity

$$\delta\xi = (J^T J)^{-1} J^T \delta u = J^+ \delta u, \quad (2)$$

hence $\|\delta\xi\| \leq \|J^+\| \|\delta u\| = \sigma_{\min}(J)^{-1} \|\delta u\|$. When J is ill-conditioned (planar targets, far range, near fronto-parallel views, or small image support), even pixel-level corner shifts induce noticeable pose errors, especially in the lateral components (t_x, t_y) .

b) *Planar markers:* For planar boards with $Z = 0$, image points follow a homography $u \sim H\tilde{X}$ where $\tilde{X} = [X \ Y \ 1]^T$ are the 2D homogeneous coordinates and $H = K[r_1 \ r_2 \ t]$ with r_1, r_2 being the first two columns of the rotation matrix R . Corner shifts Δu perturb the DLT estimate $H \rightarrow H + \Delta H$. For the translation component, the perturbation propagates as follows.

$$\delta t \approx \lambda K^{-1} \Delta h_3, \quad (3)$$

where Δh_3 is the third column of ΔH and depends linearly on Δu through the DLT solution. Thus small corner shifts directly yield errors in (t_x, t_y) , which is the basis for our geometric projection attack.

c) *Initial adversarial marker generation:* At the designated attack view, we require the estimated lateral translation to vanish, $t_x \approx 0$, $t_y \approx 0$. In the homography decomposition $H = K[r_1 \ r_2 \ t]$, this corresponds to

$$K^{-1} h_3 \parallel \mathbf{e}_z \iff h_3 \parallel K \mathbf{e}_z, \quad (4)$$

where $\mathbf{e}_z = [0, 0, 1]^T$ is the z-axis unit vector, meaning that the third column h_3 of H aligns with the optical axis direction $K \mathbf{e}_z$. Equivalently, in normalized image space, the perturbed corners form a configuration geometrically consistent with a fronto-parallel observation (i.e., the camera viewing direction is perpendicular to the marker plane), enforcing $(t_x, t_y) \approx (0, 0)$. This gives a clear geometric initialization for the attack view and, together with the sensitivity analysis above, shows why small corner shifts can induce large pose errors.

B. Adversarial Optimization Framework

Since the standard Perspective-n-Point (PnP) estimation pipeline relies on non-differentiable iterative solvers, gradient-based optimization cannot be directly applied. We therefore formulate the attack generation as a black-box optimization problem and employ CMA-ES to search the high-dimensional perturbation space.

1) **Problem Modeling and Search Space.** For a layout containing M markers, each with four corners, we define the optimization vector $\mathbf{x} \in \mathbb{R}^{8M}$. Each marker m contributes eight degrees of freedom representing the 2D planar shifts $(\Delta x, \Delta y)$ for its four corners:

$$\mathbf{x} = [\Delta x_{1,1}, \Delta y_{1,1}, \dots, \Delta x_{M,4}, \Delta y_{M,4}]^T. \quad (5)$$

The perturbed corner coordinates are recovered via $\mathbf{p}' = \mathbf{p} + \mathbf{x}$. To ensure physical feasibility and prevent inter-marker overlap, $\mathbf{x}$ is bounded strictly within layout-specific constraints $\mathcal{B}$.

2) Adversarial Objectives. We propose two distinct attack modalities to exploit the UAV’s horizontal positioning logic, categorized by their adversarial intent:

Directional Max Attack (DMA): This mode follows a “push” logic, designed to induce the maximum possible lateral drift along a hazardous direction. Let $\mathcal{P}(\cdot) \in \mathbb{R}^3$ denote the 3D translation vector estimated by the PnP solver. We extract its horizontal components using the subscript h , such that $\mathbf{T}_h = [t_x, t_y]^\top \in \mathbb{R}^2$.

By defining a normalized 2D target vector $\mathbf{v}_{\text{dir}} \in \mathbb{R}^2$, we maximize the projection of the horizontal estimation error onto this specific vector:

$$L_{\text{adv_max}} = -(\mathcal{P}(\mathbf{p} + \mathbf{x}) - \mathbf{T}_{gt})_h \cdot \mathbf{v}_{\text{dir}}, \quad (6)$$

where the term $(\mathcal{P}(\mathbf{p} + \mathbf{x}) - \mathbf{T}_{gt})_h$ directly formulates the 2D drift error, and $\mathbf{T}_{gt} \in \mathbb{R}^3$ is the ground-truth translation.

Targeted Frustum Attack (TFA): This mode uses a “pull” strategy, creating a false spatial setpoint that draws the UAV away from the intended landing point. Specifically, we define a direction-conditioned observation frustum Φ along a target hazardous azimuth. Our objective is to continuously minimize the horizontal translation estimate ($\|t_x, t_y\|_2$) output by the PnP solver whenever the marker is observed from any viewpoint within this frustum. By minimizing this error, the pose estimate makes the UAV believe it is moving toward the marker center, so it continues along the target azimuth until the marker leaves the camera view.

To ensure the attack remains robust throughout the UAV’s dynamic descent, we optimize this deception over a set of sampled viewpoints $v \in \Phi$:

$$L_{\text{adv_tgt}} = \sum_{v \in \Phi} \tilde{w}_v \|\mathcal{P}(\mathbf{p} + \mathbf{x}, v)_h\|_2, \quad (7)$$

where v represents a specific camera pose sampled from the frustum Φ , $\tilde{w}_v$ is the normalized weight for viewpoint v , and $\mathcal{P}(\mathbf{p} + \mathbf{x}, v)_h \in \mathbb{R}^2$ denotes the horizontal component of the estimated translation when observed from v . By aggregating the loss across a spatial distribution of viewpoints, this deception becomes resilient to perspective changes, providing a stable guidance effect that continuously draws the UAV toward the hazardous direction.

3) Physical Feasibility and Stealthiness Constraints. To enforce physical constraints and preserve visual stealthiness, we add penalty terms to the objective to discourage invalid geometries and visible distortions.

Visual Stealth via Multi-view SSIM: The Structural Similarity Index (SSIM) [29] is highly sensitive to structural and textural alterations, making it an ideal metric for preserving visual inconspicuousness. We pre-render a reference “clean” marker board I_{clean} across a set of viewpoints V_{ref} . For a candidate perturbation $\mathbf{x}$, we calculate the minimum SSIM across these viewpoints, $S_{\min}(\mathbf{x}) = \min_{v \in V_{\text{ref}}} \text{SSIM}(I_{\text{pert}}^v, I_{\text{clean}}^v)$. To enforce stealthiness, we apply a quadratic hinge penalty when $S_{\min}$ drops below a conservative threshold τ_s :

$$L_{\text{ssim}} = \begin{cases} \lambda_s(\tau_s - S_{\min})^2, & \text{if } S_{\min} < \tau_s \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where λ_s is a large penalty coefficient.

Planar Normal Consistency: Extreme corner shifts can lead to geometrically inconsistent 3D estimations, where markers on the same rigid board appear to have divergent orientations. For each marker m , we perform an individual PnP estimation to obtain its rotation matrix $\mathbf{R}_m$, extracting the surface normal $\mathbf{n}_m$. We penalize the maximum angular discrepancy $\theta_{\max} = \max_{m,k} \angle(\mathbf{n}_m, \mathbf{n}_k)$ if it exceeds a small angular threshold θ_{thr} :

$$L_{\text{normal}} = \begin{cases} \lambda_n(\theta_{\max} - \theta_{\text{thr}})^2, & \text{if } \theta_{\max} > \theta_{\text{thr}} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

This constraint enforces geometric reasonableness, restricting perturbations to a planar model with small deviations.

Detection Robustness Penalty: To reject degenerate solutions that break marker detectability, we incorporate a hard-penalty term L_{det} . If any evaluation step—including Joint PnP or individual marker detection—fails for a candidate $\mathbf{x}$, the loss function returns a high constant value λ_{fail} . This ensures the optimized markers remain within the operational envelope of the ArUco detector.

4) Joint Optimization Objective: The overall fitness function evaluated by the CMA-ES optimizer is a weighted sum of the adversarial loss and the physical constraints:

$$L_{\text{total}} = w_{\text{adv}} L_{\text{adv}} + L_{\text{ssim}} + L_{\text{normal}} + L_{\text{det}}, \quad (10)$$

where L_{adv} is selected based on the desired attack modality ($L_{\text{adv_max}}$ or $L_{\text{adv_tgt}}$). By minimizing L_{total} , we generate adversarial patterns that balance attack potency with practical detectability and stealth.

IV. EVALUATION AND RESULTS

A. Sensitivity Analysis of ArUco Pose Estimation

We first measure how small corner perturbations translate into pose errors in the ArUco pipeline. This sensitivity study motivates our attack design by showing how minor geometric changes can produce large pose errors.

1) Evaluation Setup: All evaluations are conducted within a high-fidelity rendering environment. The benchmark employs a standard 10 cm ArUco marker positioned at a 1.0 m frontal distance. We perform a statistical evaluation using 1,441 perturbation samples per marker, with sub-centimeter shifts up to 1.0 cm applied to the top-right corner while the other three corners remain fixed. To test generality, we evaluate three ArUco dictionaries from OpenCV, “aruco::DICT_4X4_250”, “aruco::DICT_5X5_250”, and “aruco::DICT_6X6_250”, and randomly sample 20 marker IDs from each dictionary. We report *Translation Error* and *Rotation Error*. Following prior work [30], [31], we use 5 cm and 5° as the safety-critical thresholds for high-precision pose estimation in UAV landing scenarios.

2) Results and Analysis: As illustrated in Fig. 3, our evaluation across different dictionaries reveals the following key observations:

Geometric-ID Decoupling: We observe that the pose error variance across different dictionaries and marker IDs is negligible. This confirms a structural separation in the detection

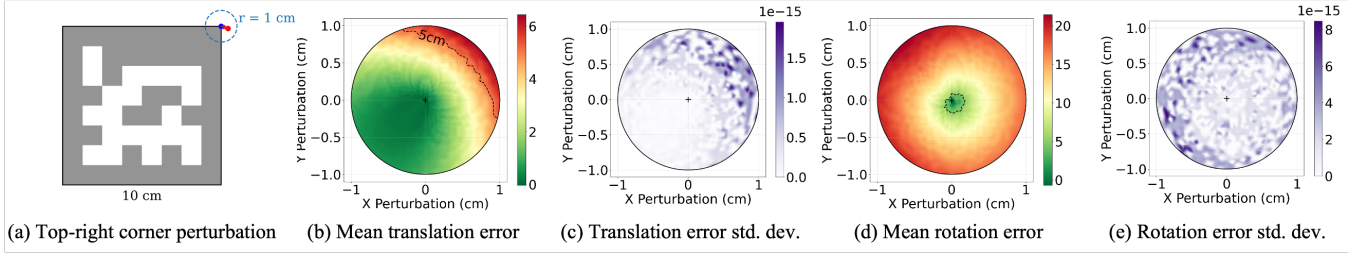


Fig. 3. Sensitivity analysis under corner perturbations using 60 markers from three OpenCV ArUco dictionaries (20 IDs each). (a) Top-right corner is perturbed within a 1 cm radius. (b) Mean translation error in centimeters. (c) Standard deviation of translation error. (d) Mean rotation error in degrees. (e) Standard deviation of rotation error in degrees.

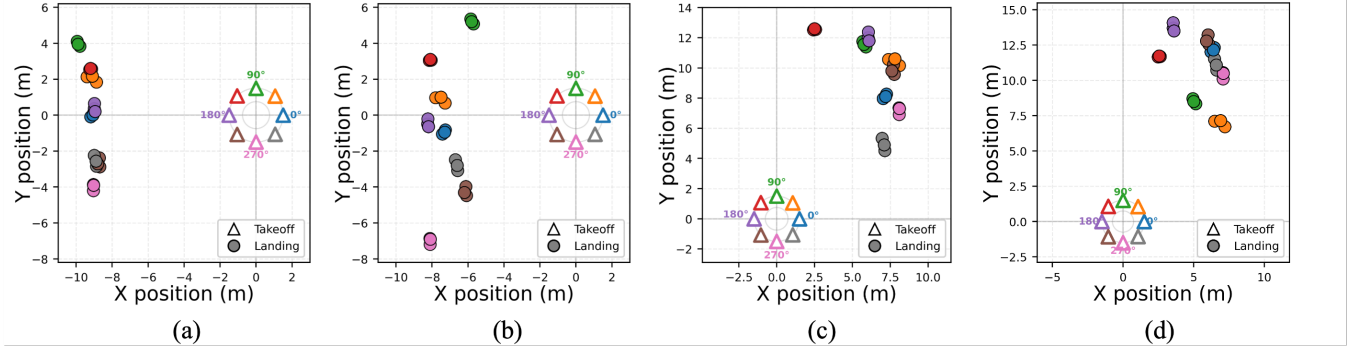


Fig. 4. Touchdown scatter plots under different attacks and board layouts. (a) Nested layout with Directional Max Attack (DMA). (b) Grid-board layout with DMA. (c) Nested layout with Targeted Frustum Attack (TFA). (d) Grid-board layout with TFA.

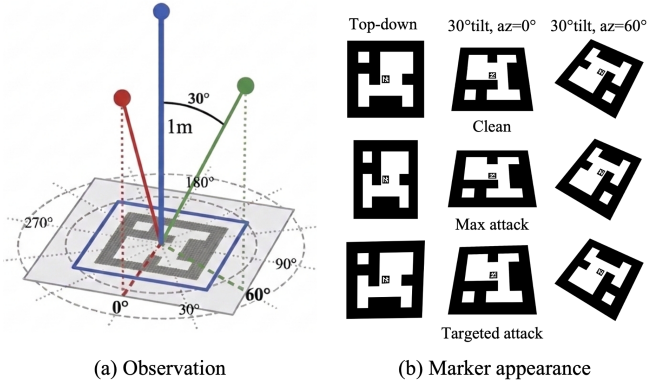


Fig. 5. Visual stealthiness of MirageMarker on the nested board. (a) Three views: top-down and two 30° tilted views at azimuths 0° and 60°, matching the three columns in (b). (b) Marker-board appearance under these views: clean (top row), DMA (middle row), and TFA (bottom row).

pipeline: The marker’s binary code pattern is used only for ID decoding and validation, whereas the 6-DoF pose is estimated solely from the 2D image coordinates of the detected corners. Consequently, corner-based geometric attacks are dictionary-agnostic and remain effective regardless of the marker’s ID pattern.

Sensitivity and Error Amplification: The results demonstrate that even a perturbation applied to a single corner induces a significant impact on pose estimation. In particular, the rotation estimate is highly sensitive: within a 1.0 cm perturbation radius, the rotation error can readily exceed 20°, far above the 5° accuracy requirement commonly assumed for precision pose estimation tasks. Translation errors are also substantial and can surpass 5.0 cm. Since this experiment perturbs only one corner at a time, optimizing all four corners

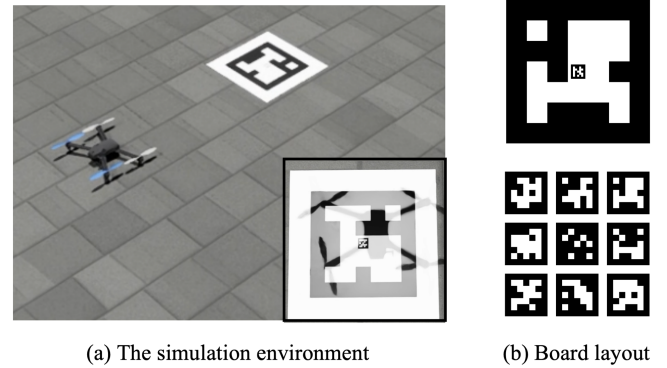


Fig. 6. Simulation setup and marker-board layout. (a) Landing scene and downward camera view. (b) Two board layouts used in our experiments: a nested board with a 50 cm outer marker and a 5 cm inner marker, and a grid board with 15 cm markers spaced 2.5 cm apart.

jointly provides a larger attack space and can induce even stronger, directionally controlled pose errors.

B. End-to-End UAV Precision Landing Evaluation

We next evaluate the system-level effects of our DMA and TFA in an autonomous precision landing task.

1) Evaluation Setup: We evaluate autonomous precision landing in a high-fidelity NVIDIA Isaac Sim environment. The UAV’s flight control follows the native `AUTO.PRECLAND` mode of the PX4 autopilot without any modification to its internal landing logic. We utilize a down-facing monocular camera (2560×1440, 90° FoV) to track two 50 cm×50 cm layouts: a 3×3 grid board and a nested board. The two layouts represent common UAV landing designs, with a distributed grid for spatial redundancy [2] and a multi-scale nested board for detection during de-

TABLE I

END-TO-END LANDING PERFORMANCE. MRE IS THE MEAN TOUCHDOWN DISTANCE TO THE BOARD CENTER (M). OUTSIDE: TOUCHDOWN OUTSIDE THE LANDING BOARD (%). $d > 5$ M: LARGE TOUCHDOWN ERROR EXCEEDING 5 M (%). TARGET LOSS: MARKER LOST DURING LANDING (%).

Layout	Method	MRE	Outside	$d > 5$ m	Target loss
Board	Clean	0.02	0	0	0
	DMA	8.12	100	100	100
	TFA	12.37	100	100	100
Nested	Clean	0.03	0	0	0
	DMA	9.48	100	100	100
	TFA	11.9	100	100	100

scent [23], [24]. All markers are placed at the same height on the landing-board surface. Fig. 6 illustrates the simulation environment and marker board layouts. We optimize both board layouts in Fig. 6(b) using our DMA and TFA and optimization details are provided in the Appendix.

Each trial begins with the UAV taking off from a position with a 1.5 m horizontal offset from the board center. The UAV takes off to 5 m from one of eight initial azimuths $\{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$ and then switches to the precision-landing mode. We repeat each (offset, azimuth) setting three times.

2) Metrics and Baselines: Landing performance is evaluated by the final touchdown distance d to the board center. We report the mean radial error (MRE) over all azimuths and a graded attack success rate (ASR) using three criteria: (i) $d_1 > 0.25\sqrt{2}$ m, the landing point outside the landing pad; (ii) $d_2 > 5.0$ m, large deviation from the landing pad; and (iii) target loss when the markers leave the camera field of view. We also show touchdown scatter plots (Fig. 4).

3) Results and Analysis: In our end-to-end evaluations, the UAV follows the biased pose feedback and keeps flying along the induced direction until the board leaves the camera view. PX4 then enters its default search behavior, climbs to the preset search altitude, and descends when the target is not reacquired; under our attacks, the drift is large enough that the board is not found during this search, so the vehicle lands far from the pad. Fig. 4 shows the resulting takeoff and touchdown locations under four settings. DMA shifts touchdowns to one side, and TFA shifts them along the target azimuth. Table I summarizes landing accuracy using the touchdown distance to the board center. Clean marker boards remain close to the pad ($\text{MRE} \leq 0.03$ m). All attack evaluations yield meter-level errors and reach 100% for *Outside*, $d > 5$ m, and *Target loss*.

4) Visual Stealthiness: As shown in Fig. 5, we assess visual stealthiness using a close-range setup with the camera placed 1.0 m from the marker center under three views: a top-down view and two 30° tilted side views at azimuths 0° and 60° . This distance is intentionally strict, since in practical deployments the marker is typically viewed from farther away and is rarely inspected at close range. Across all three views, the attacked boards remain visually close to the clean board, and the small corner perturbations are

TABLE II

RANSAC ROBUSTNESS WITH OPENCV DEFAULT THRESHOLD (8 PX). REJECT. IS THE OUTLIER REJECTION RATE; H-ERR IS THE MEAN HORIZONTAL TRANSLATION ERROR IN METERS.

Layout	Method	1.5 m		3 m		5 m	
		Reject.	H-err	Reject.	H-err	Reject.	H-err
Board	Clean	0/36	0	0/36	0	0/36	0
	DMA	11/36	0.49	4/36	1.19	5/36	2.05
	TFA	0/36	0.08	1/36	0.58	1/36	1.14
Nested	Clean	0/8	0	0/8	0	0/8	0
	DMA	2/8	0.30	2/8	0.95	2/8	2.09
	TFA	0/8	0.09	0/8	0.27	0/8	0.88

TABLE III

KF χ^2 INNOVATION-GATE PASS RATE OVER A FULL LANDING EVALUATION. PASS RATE IS COMPUTED OVER VISION UPDATES THAT REACH THE GATE, USING THE THRESHOLD $\beta \leq 3.84$.

Layout	Method	KF-pass rate (%)
Board	Clean	100
	DMA	96.6
	TFA	97.8
Nested	Clean	100
	DMA	99.3
	TFA	99.5

difficult to distinguish especially in the side views.

C. Robustness to Conventional Filtering

To evaluate robustness against standard filtering, we test whether our attacks remain effective under RANSAC [32] for spatial outlier rejection and a Kalman filter (KF) [33] with a χ^2 innovation gate for temporal gating.

1) RANSAC: We evaluate OpenCV `solvePnP` Ransac in the simulated setup using the default reprojection threshold of 8.0 px. We run the test under a top-down view at three marker-center distances of 1.5 m, 3 m, and 5 m. Table II reports the outlier rejection rate and the resulting horizontal position error $e_h = \sqrt{e_x^2 + e_y^2}$.

Table II shows that both attacks remain effective across all tested distances, and the horizontal error increases with distance. Under the default 8 px threshold, most perturbed corners are still treated as inliers, so RANSAC returns a pose estimate with a pronounced horizontal error rather than rejecting the measurement. This suggests that outlier rejection alone is insufficient for corner-shift attacks that preserve overall geometric consistency. A small rejection count is common under noisy measurements. RANSAC cannot tell whether the rejections come from noise or our attack. We also observe that DMA causes slightly higher rejection than TFA, yet the returned pose remains strongly biased. TFA is typically accepted by RANSAC with few corner rejections, especially under the nested layout where all perturbed corners are retained across distances, while the horizontal bias persists.

2) KF χ^2 innovation gate: As a robustness baseline in our end-to-end landing experiments, we follow the PX4 landing-target estimator and apply its default per-axis χ^2 innovation

gate. We reject a vision update when the normalized innovation exceeds 3.84 at 95% confidence, and follow the PX4 default settings for the remaining filter parameters.

Table III shows that the KF χ^2 gate passes nearly all vision updates during landing. The pass rate stays above 96% under both DMA and TFA for both layouts, so most attacked measurements are not rejected by innovation gating. This standard robustness check does not filter out our attack.

V. CONCLUSION & FUTURE WORK

This paper shows that ArUco-based precision landing can be compromised by small plausible corner shifts that preserve detection yet induce large pose bias. We present MirageMarker, a black-box CMA-ES framework that searches over printable corner perturbations under multi-objective constraints on feasibility and stealth. In end-to-end landing experiments with PX4's native precision-landing mode, MirageMarker produces meter-scale touchdown errors across two representative board layouts and diverse takeoff azimuths. Standard filtering baselines commonly used in practice, including RANSAC and a χ^2 innovation gate, do not remove our corner perturbations. The results highlight the need to evaluate pose reliability and system-level behavior in real-world UAV landing systems. Future work will validate MirageMarker on real UAV systems to assess its physical-world transferability and evaluate it on marker layouts with varying heights and other fiducial marker designs, including fractal markers, through corresponding geometric and pose-estimation adaptations to further investigate generalizability.

VI. ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments and suggestions, which have helped to improve the quality and presentation of the paper. This research was supported by the NSF under grant CNS-2145493, and by the NASA University Leadership Initiative under Award 80NSSC24M0070.

REFERENCES

- [1] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2016.
- [2] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, 2014.
- [3] D. Jourdan, A. Sekine, S. Velez, and A. Prager, "Adhoc geo-fiducial mats for landing uavs," U.S. Patent 11 745 899 B2, Sep. 5 2023.
- [4] "Meet the drones taking delivery to new heights," <https://wing.com/technology>.
- [5] Y. Fan, "Positioning reference device," China Patent CN108981705B, Sep. 24 2021.
- [6] "A Tech-driven Retail Company," <https://www.meituan.com/en-US/about-us>.
- [7] R. Berral-Soler, R. Muñoz-Salinas, R. Medina-Carnicer, and M. J. Marín-Jiménez, "Deeparuco: Marker detection and classification in challenging lighting conditions," in *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2023.
- [8] R. Berral-Soler, R. Muñoz-Salinas, R. Medina-Carnicer, and M. J. Marín-Jiménez, "Deeparuco++: Improved detection of square fiducial markers in challenging lighting conditions," *Image and Vision Computing*, 2024.
- [9] S. Xie, M. H. Fakihi, J. Lu, F. Alshammari, N. Wang, T. Sato, H. Bouzidi, M. A. A. Faruque, and Q. A. Chen, "Flytrap: Physical distance-pulling attack towards camera-based autonomous target tracking systems," *Network and Distributed System Security (NDSS) Symposium*, 2026.
- [10] J. Tian, B. Wang, R. Guo, Z. Wang, K. Cao, and X. Wang, "Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles," *IEEE Internet of Things Journal*, 2021.
- [11] A. Raja, L. Njilla, and J. Yuan, "Adversarial attacks and defenses toward ai-assisted uav infrastructure inspection," *IEEE Internet of Things Journal*, 2022.
- [12] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep n p: An accurate o (n) solution to the p n p problem," *International Journal of Computer Vision*, 2009.
- [13] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in *IEEE Conference on Technologies for Homeland Security (HST)*, 2012.
- [14] N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *IEEE/IFIP Network Operations and Management Symposium*, 2016.
- [15] A. Khazraei, H. Meng, and M. Pajic, "Stealthy perception-based attacks on unmanned aerial vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [16] J. Tiemann, F. Schweikowski, and C. Wietfeld, "Design of an uwb indoor-positioning system for uav navigation in gnss-denied environments," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015.
- [17] K. Choutri, M. Lagha, S. Meshoul, H. Shaiba, A. Chegrani, and M. Yahiaoui, "Vision-based uav detection and localization to indoor positioning system," *Sensors*, 2024.
- [18] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [19] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers," *Journal of Intelligent & Robotic Systems*, 2021.
- [20] A. Marut, K. Wojtowicz, and K. Falkowski, "Aruco markers pose estimation in uav landing aid system," in *IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2019.
- [21] A. Khazetdinov, A. Zakiev, T. Tsoy, M. Svinin, and E. Magid, "Embedded aruco: a novel approach for high precision uav landing," in *International Siberian Conference on Control and Communications (SIBCON)*, 2021.
- [22] M. F. Sani and G. Karimian, "Automatic navigation and landing of an indoor ar. drone quadrotor using aruco marker and inertial sensors," in *International Conference on Computer and Drone Applications (ICConDA)*, 2017.
- [23] L. Mu, S. Cao, Y. Zhang, X. Zhang, N. Feng, and Y. Zhang, "Autonomous landing guidance for quad-uavs based on visual image and altitude estimation," *Drones*, 2025.
- [24] J. Lee, S. Y. Choi, and T. Bretl, "The use of multi-scale fiducial markers to aid takeoff and landing navigation by rotorcraft," in *AIAA SCITECH 2024 Forum*, 2024.
- [25] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [26] *Parrot AR.Drone 2.0 User Guide*, <https://www.parrot.com/assets/s3fs-public/2021-09/ar.drone2.user-guide.uk.pdf>.
- [27] *OV9282 1-Megapixel Product Brief*, <https://www.ovt.com/wp-content/uploads/2024/05/OV9282-PB-v1.4-WEB.pdf>.
- [28] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, 2001.
- [29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, 2004.
- [30] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [31] M. Liu, S. Li, A. Chhatkuli, P. Truong, L. Van Gool, and F. Tombari, "One2any: One-reference 6d pose estimation for any object," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [32] K. G. Derpanis, "Overview of the ransac algorithm," *Image Rochester NY*, 2010.
- [33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

APPENDIX

VII. OPTIMIZATION SETTINGS

A. *Directional Max Attack (DMA)*

DMA uses a fixed observation altitude of 5.0 m and optimizes a directional drift objective along the x-axis. The initial geometric seed uses a nominal view with tilt 10° and azimuth 0° . CMA-ES runs for up to 800 iterations. Corner-shift bounds are 0.0125 m for the grid board. For the nested board, the outer marker is bounded by 0.025 m and the inner markers are capped at 0.01 m.

B. *Targeted Frustum Attack (TFA)*

TFA uses a fixed observation altitude of 5.0 m and optimizes over a direction-conditioned frustum centered at tilt 15° and azimuth 45° . The frustum half-angle is 10° . CMA-ES runs for up to 800 iterations for the grid board and 1000 iterations for the nested board. The population size is 60 for the grid board and 30 for the nested board. Early stopping triggers after 100 iterations without improvement. Corner-shift bounds are 0.0125 m for the grid board. For the nested board, the outer marker is bounded by 0.025 m and the inner markers are capped at 0.01 m.

C. *Physical Feasibility and Stealthiness Constraints.*

Visual and geometric constraints are enabled with an SSIM threshold of 0.95 and a normal-consistency threshold of 8° . A fixed penalty of 50.0 is returned if detection or pose estimation fails.