Assuring Safety of Vision-Based Swarm Formation Control

Chiao Hsieh[1,2], Yubin Koh[1,3], Yangge Li[1] and Sayan Mitra[1]

***Abstract*— Vision-based formation control systems are attractive because they can use inexpensive sensors and can work in GPS-denied environments. The safety assurance for such systems is challenging: the vision component's accuracy depends on the environment in complicated ways, these errors propagate through the system and lead to incorrect control actions, and there exists no formal specification for end-to-end reasoning. We address this problem and propose a technique for safety assurance of vision-based formation control: First, we propose a scheme for constructing quantizers that are consistent with vision-based perception. Next, we show how the convergence analysis of a standard quantized consensus algorithm can be adapted for the constructed quantizers. We use the recently defined notion of *perception contracts* to create error bounds on the actual vision-based perception pipeline using sampled data from different ground truth states, environments, and weather conditions. Specifically, we use a quantizer in logarithmic polar coordinates, and we show that this quantizer is suitable for the constructed perception contracts for the vision-based position estimation, where the error worsens with respect to the absolute distance between agents. We build our formation control algorithm with this nonuniform quantizer, and we prove its convergence employing an existing result for quantized consensus.**

I. Introduction

Distributed consensus, flocking, and formation control have been studied extensively, including in scenarios where the participating agents only have partial state information (see, for example [1]–[4]). With the advent of deep learning and powerful computer vision algorithms, it is now feasible for agents to use vision-based state estimation for formation control (See Figure 1). Such systems can be attractive because they do not require expensive sensors and localization systems, and also can be used in GPS-denied environments [5]–[7]. However, deep learning and vision algorithms are well-known to be fragile, which can break the correctness and safety of the end-to-end formation control system. Further, it is difficult to specify the correctness of a vision-based state estimator, which gets in the way of modular design and testing of the overall formation control system [8]. In this paper, we address these challenges and present the first end-to-end formal analysis of a vision-based formation control system.

[1]The authors are with Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Champaign, IL, USA {chsieh16, yubink2, li213, mitras}@illinois.edu

[2]The author is with Graduate School of Informatics, Kyoto University, Kyoto, Japan hsieh.chiao.7k@kyoto-u.ac.jp

[3]The author is with Department of Computer Science, Purdue University, West Lafayette, IN, USA koh22@purdue.edu

This work is supported in part by USDA National Institute of Food and Agriculture (NIFA) through the National Robotics Initiative under Grant NIFA#2021-67021-33449, NSF SHF 2008883, the Boeing Company, and the Japan Science and Technology Agency.

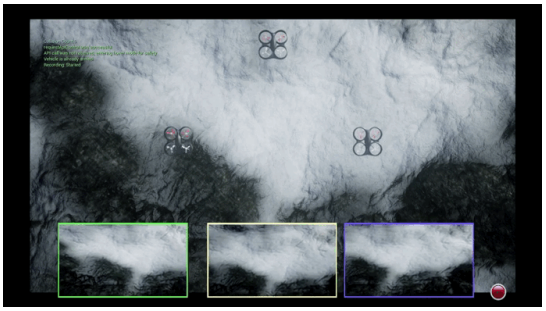


Fig. 1: Vision-based drone formation using downward facing camera images in AirSim.

We present analyses for both convergence and safety assurance of a vision-based swarm formation control system. The computer vision pipeline (See Figure 2) uses feature detection, feature matching, and geometry to estimate the relative position of the participating drones. The estimated relative poses are then used by a consensus-based formation control algorithm. There are two key challenges in analyzing the system: (1) The perception errors impact the behavior of neighboring agents and propagates through the entire swarm. (2) The magnitude of the perception error is highly nonuniform, and depends on the ground truth values of the relative position between neighboring agents. In general, perception errors can get worse as the system approaches the equilibrium (desired formation). For example, if the desired formation requires two drones to move further apart, these two drones will see less common features in camera images. This can cause less accurate estimations of relative positions, and thus make stabilization difficult. Environmental variations (e.g., lighting, fog) are other factors that can make the vision-based system unstable.

In addressing the problem, our idea is to view the vision-based formation control system as a *quantized consensus protocol* [9]. We start with the *assumption* that the impact of the state estimation errors arising from the vision pipeline can be encapsulated as quantization errors in a non-uniform quantization scheme. That is, the quantization step size can vary non-uniformly with respect to the state so that the quantization errors can overapproximate state dependent perception errors. To discharge this assumption, our analysis has to meet two requirements. First, we have to propose a *specific* quantization scheme under which the formation control system is indeed guaranteed convergence. For this, we develop a quantized formation controller (Equation (1)) and a *logarithmic polar quantizer* (Equation (2) and (3)), and we show in Theorem 1 that indeed the resulting quantized formation control protocol converges, using sufficient condi-

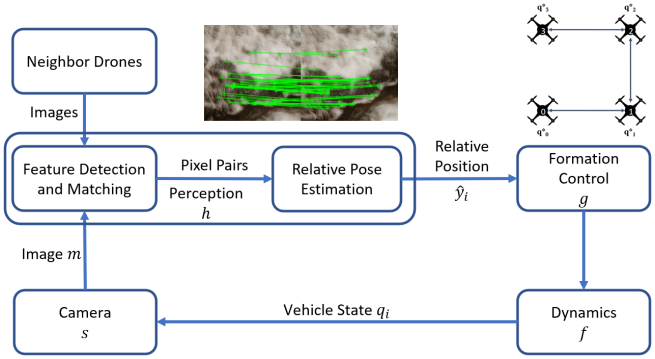


Fig. 2: Architecture of an agent in the vision-based formation control systems.

tions from [9]. Secondly, we have to show that a quantizer instantiated from the quantization scheme matches the error characteristics of the vision pipeline. For this part, we utilize the recently developed idea of *perception contracts* [10], [11]. A perception contract (PC) for a vision-based state estimator bounds the estimation error as a function of the ground truth state. Earlier in [11], PCs have been used to establish the safety of vision-based lane keeping systems. For formation control, however, the PCs are dramatically different because the error has a highly non-uniform dependency on the state; as the drones get closer, the error drops. Through data-driven construction of the logarithmic PC, we show that the vision pipeline indeed matches the PC with high probability in Section IV, and we further adapt to environmental variations by inferring quantization step sizes for different environments.

In summary, our contributions are as follows: (1) An approach to construct a quantizer as the perception contract of the vision component. (2) Empirical analysis of the impact of environmental variations on the perception contract with the photorealistic AirSim simulator [12]. (3) Theoretical analysis of the overall formation control system using the constructed quantizer, which gives the bounds on the convergence time. Our code of the vision pipeline, simulation script, and analysis tool are publicly available[1].

Related Works: Three threads of research have recently addressed formal end-to-end analysis of vision-based autonomous systems. The first thread is to search for a formal model abstracting the perception components with data driven approach, and then simplify the verification of the end-to-end system using the perception model. The works in [13]–[15] approach the problem using discrete state space and stochastic models. Our previous works [10], [11], [16] develop the idea of *perception contracts* using continuous state space models. The second thread aims at finding more tractable NNs to simplify the complex perception for formal analyses. Katz et al. [17] trains generative adversarial networks (GANs) to produce a network to simplify the image-based NN. NNLander-VeriF [18] verifies NN perception along with NN controllers for an autonomous landing system. The third thread is to extend testing with guided search and provide statistical safety guarantees. VerifAI [19] and its follow-up works uses techniques like fuzz testing on vision-based systems using simulation. It efficiently searches for synthetic camera images and scenarios where the system exhibits unsafe behavior to falsify the system specifications. Thus far all the applications studies in these threads focus on a single agent for lane following or aircraft landing with vision, which is quite different from distributed formation control. In contrast, our current work aims to provide safety analyses for a formation control system with vision-based perception, and we apply the analysis on convergence to quantized consensus [9] for safe separation and formation.

[1]Repository: https://gitlab.engr.illinois.edu/aap/airsim-vision-formation

Paper Organization: In Section II, we introduce the formation control system with the vision-based perception and review the quantized formation controller. In Section III, we show the convergence under perception error using our main theory of quantized consensus. In Section IV, we describe the quantization for perception error bounds via sampling from vision-based pose estimation with AirSim simulation. We then conclude in Section V.

II. Vision-Based Formation Control

We will study a distributed formation control system with $N+1$ identical aerial vehicles or *agents* with a leader agent 0 as shown in Figure 1. The target formation is specified in terms of relative positions between agents. Each agent i has a downward facing camera, and it uses images from its own camera and its predecessor $i-1$'s camera to periodically estimate the relative position of i with respect to $i-1$. Based on the estimated relative positions to its neighbor, agent i then updates it own position by setting a velocity, to try and achieve the target formation.

Before describing the vision and control modules in more detail, we introduce some notations. First, we describe the neighborhood relation between agents by an undirected connected graph $G=(V,E)$, where $V=\{0,1,\ldots,N\}$. Second, we only consider planar formations for simplicity though the agents are in 3-dimensional space. Thus, the position of agent i in the world frame is represented by a vector $q_i \in \mathbb{R}^2$. The state of the overall system is a sequence $\mathrm{q}=(q_0,q_1,\ldots,q_N)$. The distributed formation control system evolves with a goal of reaching a target formation in a set $\mathrm{E}_{\mathrm{q}^*}$ that is specified by a vector $\mathrm{q}^*=(q_0^*,q_1^*,\ldots,q_N^*)$ as

$$\mathrm{E}_{\mathrm{q}^*}=\{\mathrm{q} \mid \forall i \in \{1,\ldots,N\}, q_i-q_{i-1}=q_i^*-q_{i-1}^*\}.$$

That is, $\mathrm{E}_{\mathrm{q}^*}$ is the set of all states that form q^* up to translations. We also specify a *safe set* that the where the distance between two agents is never too close or too far:

$$\mathrm{S}_{\mathrm{q}}=\{\mathrm{q} \mid \forall i \in \{1,\ldots,N\}, d_{\min}<\|q_i-q_{i-1}\|<d_{\max}\}$$

where $0<d_{\min}<d_{\max}$ defines the range of safe distances.

A. Vision-Based Relative Pose Estimation

We now discuss the components of an agent i (Figure 2). Agent i's downward-facing camera s periodically generates an image of the ground m_i, which depends on its state q_i and other environmental factors like background scenery,

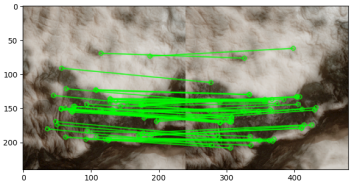


Fig. 3: Feature matching on an image pair from AirSim.

lighting, fog, etc. The neighboring agent $i-1$ generates another image m_{i-1} of the ground and shares this with agent i over the communication channel. We assume the whole system runs synchronously in lock-step, i.e., both neighboring drones will capture the image at the same time and there's no communication delay between drones while sharing images. The vision-based pose estimation algorithm h takes a pair of images, m_{i-1} and m_i, as an input and produces the *estimated relative position* $\hat{y}_i$ to estimate the relative position of agent i with respect to agent $i-1$, i.e., $q_i - q_{i-1}$. The estimation algorithm in general follows these steps: (1) First, h detects features from each image. Any of the various feature detection algorithms like SIFT [20], SURF [21], and ORB [22] can be used for this step. (2) Then, h collects the detected features from the pair of images, and a feature matching algorithm (such as FLANN [23]) is used to match pairs of features in each image as shown in Figure 3. (3) For each feature point, the relationship between the pixel coordinate and the world coordinate of the feature point [24] is described by $s[u\ v\ 1]^T = K[R \mid t][X\ Y\ Z\ 1]^T$ where $[u\ v\ 1]^T$ is the pixel coordinate, $[X\ Y\ Z\ 1]^T$ is the world coordinate of each detected feature, K is the camera intrinsic matrix and $[R \mid t]$ is the extrinsic camera parameters. With a set of at least 8 matched features, we can come up with 8 pairs of equations between the poses of two cameras, and by solving these equations, we can calculate the relative rotation and the normalized translation vector using the inverse geometry of image formation. Examples of this step appear in [25]–[27]. Further, the altitude information and drone orientation can be used to estimate the true distance to ground and recover the length of the translation vector.

The accuracy of the perception pipeline can be influenced by many factors. The change of environments such as background, lighting, and weather influence the quality of the image and the image features, which in turn influence the accuracy of relative pose estimation.

B. Formation Control in Relative Positions

To simplify the notations, let Y denote the vector space of the *relative positions* between pairs of drones. Let $y_i \in Y$ be defined as $y_i = q_i - q_{i-1}$, for $i = 1,\dots,N$. A state is a sequence of relative positions $\mathrm{y} = (y_1, y_2, \dots, y_N)$. Let $y_i^* \in Y$ be the desired relative vector between drone $i-1$ and i, i.e., $y_i^* = q_i^* - q_{i-1}^*$. The target equilibrium state is defined by:

$$\mathrm{y}^* = (y_1^*, \dots, y_N^*),$$

and the safe set is

$$\mathrm{S_y} := \{\mathrm{y} \mid \forall i \in \{1,\dots,N\}, d_{\min} < \|y_i\| < d_{\max}\}.$$

The following proposition relating the y-system with the original q-system follows immediately.

Proposition 1. *The formation control system (*q*-system) reaches a desired state in* $\mathrm{E_{q^*}}$ *if and only if the system in relative positions (*y*-system) reaches the target* y^**. In addition, the formation control system (*q*-system) stays within the safe set* $\mathrm{S_q}$ *if and only if the system in relative positions (*y*-system) stays within the safe set* $\mathrm{S_y}$.

C. Quantized Formation and Perception Contract

Given the system in relative positions with the state at time t represented by $\mathrm{y}[t] = (y_1[t], \dots, y_N[t])$, we aim to handle perception errors by designing a *quantizer* Q and a quantized formation controller built on top of Q. The insight is as follows: If the ground-truth $y_i[t]$ and the perceived relative position $\hat{y}_i[t]$ always lead to the same quantized value after quantization, i.e., $Q(y_i[t]) = Q(\hat{y}_i[t])$, then any quantized formation controller using $Q(\hat{y}_i[t])$ instead of $Q(y_i[t])$ will still stabilize the system regardless of perception errors.

More precisely, we follow the definitions in [28] and choose a subset of positions $W = \{w_1, w_2, \dots\} \subset Y$ selected for quantization. A *quantizer* is a function $Q : Y \to W$ which partitions Y into *quantization regions* of the form $\{y \in W \mid Q(y) = w_k\}$ for each $w_k \in W$. Given a target state $\mathrm{y}^* = (y_1^*, \dots, y_N^*)$ with all $y_i^* \in W$, a *quantized formation controller* ensures that the system is evolving between states with all $y_i[t] \in W$ for all time t, and stabilizes the system to the target y^*. The *perception contract* (*PC*) then requires that the perceived value is always in the quantization region defined by the ground-truth quantized value, that is, $\hat{y}_i[t] \in \{y \mid Q(y) = y_i[t]\}$ or equivalently $Q(\hat{y}_i[t]) = Q(y_i[t])$.

In this paper, we study a particular template of quantized formation controllers constructed using a quantizer, a difference function, and a weighted average function. We assume a generalized difference function *diff* to calculate the difference between two relative positions and a function $mean_\omega$ to compute a weighted midpoint parametrized by a positive real value ω.[2] The system evolves according to a (nondeterministic) discrete dynamical system with a pair (i, j) selected randomly at each time step t, and the quantized formation controller updates the pair of states as follows:

$$\begin{aligned} y_i[t+1] &= Q(mean_\omega(Q(y_i[t]), diff(Q(y_j[t]), diff(y_j^*, y_i^*)))) \\ y_j[t+1] &= Q(mean_\omega(Q(y_j[t]), diff(Q(y_i[t]), diff(y_i^*, y_j^*)))) \end{aligned} \tag{1}$$

Note that, by design, the updated states are always quantized values given any *diff* and $mean_\omega$, and they are unaffected by perception errors whenever the perception contract holds.

This template allows us to design a quantized formation controller with vision-based perception in two separate steps: (1) find sufficient conditions for the quantizer that ensures the convergence and safety of the quantized formation controller and (2) construct the quantizer from observed perception errors to serve as the perception contract. In Section III, we derive the sufficient conditions for the safety and convergence of the quantized formation. We prove that, given a *quantizer* over *logarithmic polar* (log-polar) coordinates,

[2]We will provide the criteria for selecting a proper ω in Section III-B which are required by the quantized averaging algorithm from [9].

the system in Equation (1) safely converges to the target state, and the expected value for the time of convergence is bounded. In Section IV, we study the empirically observed perception error, and we explain how to design a quantizer to approximate the perception error.

III. Convergence of Quantized Formation

In this section, we prove that the true relative positions between agents, with a quantizer on log-polar coordinates modeling the perception error, converges to the target formation and stays within the safe distance bounds. This is done by proving that the system in Equation (1) is simulated by a *quantized averaging* algorithm in [9], which is proven to always converge to quantized consensus and stay within a bounded interval. In addition, the expected convergence time of the algorithm is bounded. We first provide the quantizer in Section III-A. We then prove the simulation relation between formation control systems and quantized averaging algorithms in Section III-B, and the bound on the convergence time in Section III-C.

A. Quantization on Logarithmic Polar Coordinates

We first define the set of selected positions W in polar coordinates. Without loss of generality, we choose a quantization *step radius* $a > 1$ and define the set of quantized radii $R = \{a^0, a^{\pm 1}, a^{\pm 2}, \dots\}$ and a *step angle* $\theta_b = \frac{2\pi}{M}$ to define the set of quantized angles $\Theta = \{0, \theta_b, \dots, (M-1)\cdot\theta_b\}$ with an integer $M \geq 2$; then we define the set of selected points W by $W = R \times \Theta$. Equivalently, given two positions in polar coordinates $y_i = [r_i\ \theta_i]^T$ and $y_j = [r_j\ \theta_j]^T$ with angles normalized to $[0, 2\pi)$, we define the quantizer and other functions for the radial coordinate as below:

$$\begin{aligned} I(r_i) = \lfloor \log_a r_i \rceil; \quad Q(r_i) = a^{I(r_i)}; \\ \mathit{diff}(r_i, r_j) = \frac{r_i}{r_j}; \quad \mathit{mean}_\omega(r_i, r_j) = r_i^{(1-\omega)} \cdot r_j^{\omega} \end{aligned} \tag{2}$$

where $\lfloor \cdot \rceil$ rounds the real number to the nearest integer. The functions for the angular coordinate are as below:

$$\begin{aligned} I(\theta_i) \equiv_M \left\lfloor \frac{\theta_i}{\theta_b} \right\rceil; \quad Q(\theta_i) = I(\theta_i) \cdot \theta_b; \\ \mathit{diff}(\theta_i, \theta_j) = \theta_i \oplus -\theta_j; \\ \mathit{mean}_\omega(\theta_i, \theta_j) = \theta_i \oplus (\omega \cdot (\theta_j \oplus -\theta_i)) \end{aligned} \tag{3}$$

where $\equiv_M$ means congruent modulo M, $\oplus$ is the addition in the commutative group $SO(2)$, and $-\theta_i$ represents the additive inverse of θ_i. The mean_ω function calculates the weighted geometric mean of rotations [29].

B. Simulation by Quantized Averaging Algorithms

Following [9], let $\mathrm{z}[t] = (z_1[t], \dots, z_N[t])$ denote the state in the integer domain at each time step. An instance of quantized averaging algorithms evolves as follows. At each time step, if a pair (i, j) is randomly selected at time t, we update the values of z_i and z_j as the Equation (4) below:

$$\begin{aligned} z_i[t+1] = z_i[t] + \left\lfloor \omega \cdot (-z_i[t] + z_j[t]) \right\rceil \\ z_j[t+1] = z_j[t] + \left\lfloor \omega \cdot (z_i[t] - z_j[t]) \right\rceil \end{aligned} \tag{4}$$

where $\omega \in (\frac{1}{2}, \frac{3}{4})$ must not be a rational number with an even denominator [9, Algorithm 2]. The following proposition is directly from [9, Section 5].

Proposition 2. *The system in Equation* (4) *always converges to the set of equilibria:*

$$\xi = \{(z_1, \dots, z_N) \mid z_i \in \{L, L+1\}, i \in \{1 \dots N\} \sum_{i=1}^{N} z_i = S\}$$

where $S = \sum_{i=1}^{N} z_i[0]$ *is the initial sum of the system and* $L = \lfloor \frac{S}{N} \rfloor$ *is the quantized average.*

We now prove the convergence of the formation control system using Proposition 2.

Theorem 1. *For any initial state* $\mathrm{y}[0] \in \mathrm{Y}_0$ *where*

$$\mathrm{Y}_0 = \left\{ \left(\begin{bmatrix} r_1 \\ \theta_1 \end{bmatrix}, \dots, \begin{bmatrix} r_N \\ \theta_N \end{bmatrix} \right) \middle| \prod_i Q(r_i) = \prod_i r_i^* \wedge \bigoplus_i Q(\theta_i) = \bigoplus_i \theta_i^* \right\},$$

the formation control system in Equation (1) *converges to the target* $\mathrm{y}^* = (y_1^*, \dots, y_N^*)$ *with* $y_i^* \in W$ *for all* i.

Proof. The proof is to show that the formation control system in Equation (1) is simulated by the quantized averaging system in Equation (4) for every time step, and thus we guarantee the convergence using Proposition 2. Recall in Section III-A that the quantizer Q has a corresponding indexing function I. Let $y_i[t] = [r_i[t]\ \theta_i[t]]^T$ and $y_i^* = [r_i^*\ \theta_i^*]^T$, we denote $z_i[t]_r$ for the radial coordinate and $z_i[t]_\theta$ for the angular coordinate, and we relate the two systems by $z_i[t]_r = I(r_i[t]) - I(r_i^*)$ and $z_i[t]_\theta = I(\theta_i[t]) - I(\theta_i^*)$ for all i.

We first derive for the radial coordinates. Given the relation $z_i[t] = I(y_i[t]) - I(y_i^*)$ for all i, we apply the indexing function I on the new state $r_i[t+1]$ to calculate $z_i[t+1]_r$.

$$\begin{aligned} I(r_i[t+1]) &= I(Q(\mathit{mean}_\omega(Q(r_i[t]), \mathit{diff}(Q(r_j[t]), \mathit{diff}(r_j^*, r_i^*))))) \\ &= \left\lfloor \log_a \left(Q(r_i[t])^{(1-\omega)} \cdot \left(\frac{Q(r_j[t]) \cdot r_i^*}{r_j^*}\right)^{\omega} \right) \right\rceil \\ &= \lfloor (1-\omega) \cdot (\log_a Q(r_i[t])) \\ &\quad + \omega \cdot (\log_a Q(r_j[t]) - \log_a r_j^* + \log_a r_i^*) \rceil \\ &\quad \because r_i^*, r_j^* \text{ are quantized values.} \\ &= \lfloor (1-\omega) \cdot I(r_i[t]) \ + \omega \cdot (I(r_j[t]) - I(r_j^*) + I(r_i^*)) \rceil \\ &= I(r_i[t]) + \lfloor -\omega \cdot (I(r_i[t]) - I(r_i^*)) + \omega \cdot (I(r_j[t]) - I(r_j^*)) \rceil \end{aligned}$$

We now calculate $z_i[t+1]_r$ as follows:

$$\begin{aligned} z_i[t+1]_r &= I(r_i[t+1]) - I(r_i^*) \\ &= I(r_i[t]) - I(r_i^*) \\ &\quad + \lfloor -\omega \cdot (I(r_i[t]) - I(r_i^*)) + \omega \cdot (I(r_j[t]) - I(r_j^*)) \rceil \\ &= z_i[t]_r + \left\lfloor \omega \cdot (-z_i[t]_r + z_j[t]_r) \right\rceil \end{aligned}$$

This is exactly the same as the quantized averaging algorithm in Equation (4). Therefore, we have shown that the indices of the quantized radial coordinates evolve according to the quantized averaging algorithm.

Similarly, we derive for the angular coordinate. Recall the definitions in Equation (3). Given two quantized angles $\theta_i = Q(\theta_i)$ and $\theta_j = Q(\theta_j)$, we can distribute the indexing

function over the addition and inverse as follows:

$$I(\theta_i \oplus \theta_j) \equiv_M \left\lfloor \frac{(I(\theta_i)+I(\theta_j))\cdot\theta_b}{\theta_b} \right\rceil \equiv_M I(\theta_i)+I(\theta_j)$$
$$I(-\theta_i) \equiv_M I(-Q(\theta_i)) \equiv_M \left\lfloor \frac{-I(\theta_i)\cdot\theta_b}{\theta_b} \right\rceil \equiv_M -I(\theta_i)$$

Additionally, when $\theta_i = Q(\theta_i)$, we can derive:

$$I(\omega\cdot\theta_i) \equiv_M \left\lfloor \frac{\omega\cdot I(\theta_i)\cdot\theta_b}{\theta_b} \right\rceil \equiv_M \lfloor \omega\cdot I(\theta_i) \rceil$$

We now calculate the index of the new state $\theta_i[t+1]$.

$$\begin{aligned} I(\theta_i[t+1]) &= I(Q(mean_\omega(Q(\theta_i[t]), diff(Q(\theta_j[t]), diff(\theta_j^*, \theta_i^*))))) \\ &\equiv_M I(Q(\theta_i[t]) \oplus (\omega\cdot(Q(\theta_j[t]) \oplus -\theta_j^* \oplus \theta_i^* \oplus -Q(\theta_i[t])))) \\ &\equiv_M I(\theta_i[t]) + I(\omega\cdot(-Q(\theta_i[t]) \oplus \theta_i^* \oplus Q(\theta_j[t]) \oplus -\theta_j^*))) \\ &\equiv_M I(\theta_i[t]) + \lfloor \omega\cdot(-I(\theta_i[t]) + I(\theta_i^*) + I(\theta_j[t]) - I(\theta_j^*)) \rceil \end{aligned}$$

Then, we calculate $z_i[t+1]_\theta$:

$$\begin{aligned} z_i[t+1]_\theta &= I(\theta_i[t+1]) - I(\theta_i^*) \\ &\equiv_M I(\theta_i[t]) - I(\theta_i^*) \\ &\quad + \lfloor \omega\cdot(-I(\theta_i[t]) + I(\theta_i^*) + I(\theta_j[t]) - I(\theta_j^*)) \rceil \\ &\equiv_M z_i[t]_\theta + \lfloor \omega\cdot(-z_i[t]_\theta + z_j[t]_\theta) \rceil \end{aligned}$$

Again, the index of the angular coordinate evolves exactly following the quantized averaging algorithm in Equation (4). The derivation for $z_j[t+1]$ is the same and skipped.

Furthermore, for any initial state $\mathrm{y}[0] \in \mathrm{Y}_0$, we can derive the initial sum of the quantized averaging system as follows:

$$\begin{aligned} &\prod_i Q(r_i[0]) = \prod_i r_i^* \ \wedge\ \bigoplus_i Q(\theta_i[0]) = \bigoplus_i \theta_i^* \\ \Rightarrow &\prod_i \frac{Q(r_i[0])}{r_i^*} = 1 \ \wedge\ \bigoplus_i (Q(\theta_i[0]) \oplus -\theta_i^*) = 0 \\ \Rightarrow &\sum_i \log_a Q(r_i[0]) - \log_a r_i^* = 0 \ \wedge\ \sum_i \frac{Q(\theta_i[0])}{\theta_b} - \frac{\theta_i^*}{\theta_b} = 0 \\ \Rightarrow &\sum_i I(r_i[0]) - I(r_i^*) = 0 \ \wedge\ \sum_i I(\theta_i[0]) - I(\theta_i^*) = 0 \\ \Rightarrow &\sum_i z_i[0]_r = 0 \ \wedge\ \sum_i z_i[0]_\theta = 0 \end{aligned}$$

As a result, the initial sum S in the z-system is 0, the quantized average L is always 0, and according to Proposition 2 the z-system converges to the only equilibrium where all $z_i = L = 0$. Therefore, y_i converges to y_i^* for all $i \in \{1,\dots,N\}$. Thus, our formation control system converges to y^*. □

The initial condition $\mathrm{y}[0] \in \mathrm{Y}_0$ states that, given the target y^*, the set Y_0 defines those initial states that will converge to y^*, so we know the system cannot go from $\mathrm{y}[0]$ to any target formation y^*. We explain this condition in Example 1 from a more practical view: Starting from an initial state $\mathrm{y}[0]$, how many target formations y^* can the system converge to?

Example 1. In this example, we simplify the system so that all agents are moving along a straight line, and hence we focus on the radial coordinates and ignore the angular coordinates. Here we consider a system with three agents, so the state of relative positions $\mathrm{y}[t] = (r_1[t], r_2[t])$ denotes the evolution of distances between agents. We assume a quantizer defined by $a = 1.2$, i.e., $Q(r_i) = 1.2^{\lfloor \log_{1.2} r_i \rceil}$. Let us start from an initial formation $\mathrm{y}[0] = (r_1[0], r_2[0]) = (0.8, 2.5)$ where the first two agents are closer and the third agent is farther. We can compute below:

$$\begin{aligned} Q(r_1[0]) = Q(0.8) &= 1.2^{\lfloor -1.22\ldots \rceil} = 1.2^{-1} \\ Q(r_2[0]) = Q(2.5) &= 1.2^{\lfloor 5.02\ldots \rceil} = 1.2^{5} \\ \prod_i r_i^* = Q(0.8) * Q(2.5) &= 1.2^{-1+5} = 1.2^4 \end{aligned}$$

Further, any target formation are of quantized values, i.e., some integer exponentiation of 1.2. W.l.o.g, we consider $\mathrm{y}^* = (1.2^m, 1.2^n)$ where $m, n \in \mathbb{Z}$. If we pick r_1 and r_2 in our first timestep and evolve according to Equation (1), we can derive:

$$\begin{aligned} r_1[1] &= Q(Q(r_1[0])^{1-\omega}(\frac{Q(r_2[0])}{r_1^*/r_2^*})^\omega) = Q(1.2^{-(1-\omega)}(\frac{1.2^5}{1.2^{m-n}})^\omega) \\ &= 1.2^{\lfloor -1+\omega(5-(-1)-(m-n)) \rceil} = 1.2^{-1+\lfloor \omega(6-(m-n)) \rceil} \\ r_2[1] &= Q(Q(r_2[0])^{1-\omega}(\frac{Q(r_1[0])}{r_2^*/r_1^*})^\omega) = Q(1.2^{5(1-\omega)}(\frac{1.2^{-1}}{1.2^{n-m}})^\omega) \\ &= 1.2^{\lfloor 5-\omega(5-(-1)-(m-n)) \rceil} = 1.2^{5+\lfloor -\omega(6-(m-n)) \rceil} \end{aligned}$$

Ensured by the criteria for selecting ω in [9, Algorithm 2], we know $\lfloor -\omega(6-(m-n)) \rceil = -\lfloor \omega(6-(m-n)) \rceil$, we then see $r_1[1]\cdot r_2[1] = 1.2^{-1}\cdot 1.2^5 = Q(r_1[0])\cdot Q(r_2[0])$ irrespective of m and n. In general, the product $\prod_i Q(r_i[0])$ is preserved in every timestep, and this applies to the target formation y^* as well, which is stated as the condition $\prod_i Q(r_i[0]) = \prod_i r_i^*$. Nevertheless, there can still be infinitely many valid target formations y^* in this example defined by integers m and n as long as $m+n=4$.

Following Example 1, we further need to integrate the safety requirement that bounds the distance between agents. Otherwise, it is unrealistic that the distance between agents can converge to arbitrarily close ($m, n \ll 0$) or far ($m, n \gg 0$). We provide the initial condition such that the system will remain in the safe set in Theorem 2.

Theorem 2. *Given a target state in the safe set* $\mathrm{y}^* \in \mathrm{S_y}$*, if the system starts in an initial state* $\mathrm{y}[0] \in (\mathrm{Y}_0 \cap \mathrm{S}_0)$ *where*

$$\mathrm{S}_0 = \left\{ (\begin{bmatrix} r_1 \\ \theta_1 \end{bmatrix}, \dots, \begin{bmatrix} r_N \\ \theta_N \end{bmatrix}) \middle| \bigwedge_{i=1}^N \frac{Q(d_{\min})}{\min_j\{r_j^*\}} < \frac{Q(r_i)}{r_i^*} < \frac{Q(d_{\max})}{\max_j\{r_j^*\}} \right\},$$

then the system will always stay in the safe set, i.e., $d_{\min} < r_i[t] < d_{\max}$ *for all time* t.

Proof. Here we show the proof steps for the minimum safe distance $d_{\min}$. We first derive the lower bound for $z_i[0]_r$ from the initial condition $\mathrm{y}[0] \in (\mathrm{Y}_0 \cap \mathrm{S}_0)$.

$$\begin{aligned} &\frac{Q(d_{\min})}{\min_j\{r_j^*\}} < \frac{Q(r_i[0])}{r_i^*} \\ \Rightarrow\ &\lfloor \log_a d_{\min} \rceil - \min_j\{\log_a r_j^*\} < \lfloor \log_a r_i[0] \rceil - \log_a r_i^* \\ \Rightarrow\ &I(d_{\min}) - \min_j\{I(r_j^*)\} < I(r_i[0]) - I(r_i^*) = z_i[0]_r \end{aligned}$$

Then, from [9, Theorem 2], we know $z_i[t]_r$ is bounded by the minimum initial value, i.e., $\min_j\{z_j[0]_r\} \le z_i[t]_r$ for all time t. We use it to derive the lower bound on $r_i[t]$ for all time t as follows.

$$\begin{aligned}
&I(d_{\min}) - \min_j\{I(r_j^*)\} < \min_j\{z_j[0]_r\} \le z_i[t]_r \\
\Rightarrow &I(d_{\min}) - \min_j\{I(r_j^*)\} < I(r_i[t]) - I(r_i^*) \\
\Rightarrow &I(d_{\min}) + (I(r_i^*) - \min_j\{I(r_j^*)\}) < I(r_i[t]) \\
\Rightarrow &I(d_{\min}) < I(r_i[t]) \qquad \because I(r_i^*) \ge \min_j\{I(r_j^*)\} \\
\Rightarrow &Q(d_{\min}) < Q(r_i[t]) \Rightarrow d_{\min} < r_i[t]
\end{aligned}$$

We skip the dual proof for the upper bound $r_i[t] < d_{\max}$. □

Remark 1. Theorem 2 suggests that we should carefully design the closest and farthest distances, $\min_j r_j^*$ and $\max_j r_j^*$, in the target state because they constrain the set of safe initial states. For example, when the distance in the target state r_j^* is close to the minimum safe distances $d_{\min}$, then the bound becomes tighter, and hence fewer initial states can ensure safety. Following Example 1, given safe distances $d_{\min} = 0.5$ and $d_{\max} = 5$, an unbalanced target state $\mathrm{y}^* = (1.2^{-2}, 1.2^6) \approx (0.7, 3.0)$ leads to a tighter bound $1.2^{-2} < \frac{Q(r_i)}{r_i^*} < 1.2^3$, and a balanced target state $\mathrm{y}^* = (1.2^2, 1.2^2) = (1.44, 1.44)$ leads to a looser bound $1.2^{-6} < \frac{Q(r_i)}{r_i^*} < 1.2^7$. In addition, Q cannot be too coarse, that is, the value of the step radius a should not be too large. Otherwise, S_0 can become an empty set, and no initial states can ensure safety.

C. Bound on Expected Convergence Time

Now, we analyze the upper bound on the convergence time. We first provide the existing result on the convergence time of quantized consensus algorithms. Then, we derive the upper bound for the formation control system.

Following [9], the probability distribution of the convergence time is defined as $T_{con}(\mathrm{z}) = \inf\{t \mid \mathrm{z}[t] \in \xi, \mathrm{z}[0] = \mathrm{z}\}$. Since our z-system evolves by a quantized gossip algorithm over *linear networks*, the upper bound on the expected convergence time is provided in [9, Lemma 7] as:

$$\max_{\mathrm{z} \in \mathrm{Z}_0} \mathbb{E}[T_{con}(\mathrm{z})] \le \frac{(z_{\max} - z_{\min})^2}{8} \cdot \frac{N(N^2-1)(N-1)}{4} \tag{5}$$

where $z_{\min}$ and $z_{\max}$ are parameters specifying the minimum and maximum integer values among all possible states, and $\mathrm{Z}_0 = \{(z_1, \ldots, z_N) \mid z_{\min} \le z_i \le z_{\max}\}$ includes all possible initial states. With the upper bound in Equation (5), we derive the bound on the expected convergence time of our formation control system.

Theorem 3. *The formation control system in Equation* (1) *converges with the following upper bound on the expected convergence time:*

$$\max_{\mathrm{y} \in (\mathrm{Y}_0 \cap \mathrm{S}_0)} \mathbb{E}[T_{con}(\mathrm{y})] \le \frac{\Delta^2}{8} \cdot \frac{N(N^2-1)(N-1)}{4}$$

where $\Delta = \max(I(d_{\max}) - I(d_{\min}), M-1)$, Y_0 *is the same in Theorem 1, and* S_0 *is the same in Theorem 2.*

Proof. The proof is to apply Equation 5 and select Δ to be the larger upper bound among the bounds on $z_{\max} - z_{\min}$ for the radial and angular coordinates in the integer domain. For the radial coordinate, we derive a lower bound for $z_{\min} = \min_i\{I(r_i) - I(r_i^*)\}$ from $\mathrm{y} \in \mathrm{S}_0$ as follows:

$$\begin{aligned}
&\bigwedge_{i=1}^{N} \frac{Q(d_{\min})}{\min_j\{r_j^*\}} < \frac{Q(r_i)}{r_i^*} \;\Rightarrow\; \frac{Q(d_{\min})}{\min_j\{r_j^*\}} < \min_i\{\frac{Q(r_i)}{r_i^*}\} \\
\Rightarrow\; &I(d_{\min}) - \min_j\{I(r_j^*)\} < \min_i\{I(r_i) - I(r_i^*)\} = z_{\min}
\end{aligned}$$

Dually, we find the bound $z_{\max} < I(d_{\max}) - \max_j\{I(r_j^*)\}$. Using the fact that $\max_j\{I(r_j^*)\} \ge \min_j\{I(r_j^*)\}$, this leads to $z_{\max} - z_{\min} < I(d_{\max}) - I(d_{\min})$.

For the angular coordinate, the value of $I(\theta_i) - I(\theta_i^*)$ is bounded by 0 and $M-1$ because of the modulo operator, hence the upper bound on $z_{\max} - z_{\min}$ is $M-1$. □

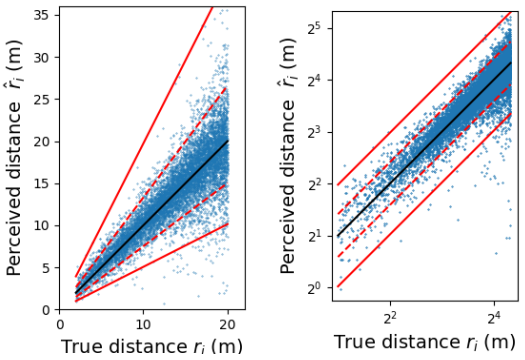
IV. Quantization as Perception Contracts

For vision-based perception, uniform worst case bounds on the perception error between the ground truth and the perceived value can be overly conservative for system-level analysis. Recent research has shown that state-dependent error models can strike a balance between the conservatism of the safety analysis and the precision of characterizing deep learning-based perception systems [10].

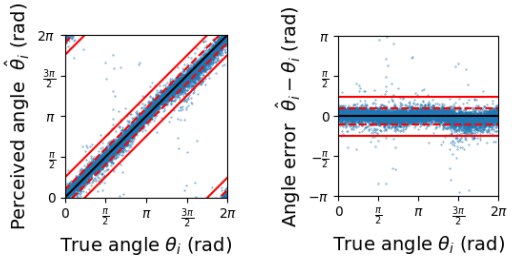
Following the same insight, we investigate the relationship between the ground truth and the perceived relative positions, and we study how to search for the parameter values for the quantizer according to the empirically observed perception errors. We randomly sampled pairs of camera images from two drones under different relative positions in AirSim. We fixed drone $i-1$ as the origin and uniformly sampled 10,000 positions of drone i within a radius between 2 m to 20 m in the AirSimNH environment from AirSim. For each sample, we obtained a pair of true relative position y_i from AirSim and perceived relative position $\hat{y}_i$ via vision-based pose estimation pipeline (of Section II-A).

Figure 4 plots the perceived position $\hat{y}_i = [\hat{r}_i\, \hat{\theta}_i]^T$ with respect to the true position $y_i = [r_i\, \theta_i]^T$. In Figure 4a, we observe that the perceived distances $\hat{r}_i$ scatter wider when the true distance r_i increases. Especially, the distribution of blue dots spreads much wider and deviates greatly from the black line, the ideal perception with no error, when the true distance crosses a certain threshold, e.g., about 15 meters in Figure 4a. This is not too surprising: As the two drones become farther apart, the overlap of the two camera views is smaller and leads to fewer matched features. Once there are less than eight pairs of matched features, it causes a sudden drop in accuracy in relative pose estimation.

Our ultimate goal is to design a quantizer whose quantization error overapproximates the perception error of the vision component, and we empirically approximate the perception error from collected samples. Recall in Section II-C, the PC enforced by the quantizer Q is $Q(\hat{y}_i) = Q(y_i)$. We further simplify the PC as $Q(\hat{y}_i) = y_i$ because y_i is a quantized value. We start by expanding the definitions in Equation 2 for the



(a) Radii on a linear scale. (b) Radii on a log scale.



(c) Angles in $[0, 2\pi)$. (d) Angles in $[0, 2\pi)$ and angle errors in $[-\pi, \pi)$.

Fig. 4: Perceived relative positions $\hat{y}_i = [\hat{r}_i \, \hat{\theta}_i]^T$ with respect to true relative position $y_i = [r_i \, \theta_i]^T$. Sampled data are blue dots. The perceived value with no error is the **black line**. The empirical bounds are plotted as red lines such that 99% of dots fall within the solid lines and 90% of the dots fall within the dashed lines.

radial coordinates. The PC can be rewritten as the following:

$$
\begin{aligned}
& Q(\hat{r}_i) = r_i \Leftrightarrow a^{\lfloor \log_a \hat{r}_i \rceil} = r_i \Leftrightarrow \lfloor \log_a \hat{r}_i \rceil = \log_a r_i \\
\Leftrightarrow \; & \log_a r_i - 0.5 \leq \log_a \hat{r}_i < \log_a r_i + 0.5 \\
\Leftrightarrow \; & \log r_i - 0.5 \cdot \log a \leq \log \hat{r}_i < \log r_i + 0.5 \cdot \log a
\end{aligned}
$$

This says that the PC defines a pair of *linear bounds* around the ground truth on a log scale. We can decrease or increase the value of a to make the PC more strict or relaxed. Ideally, the PC should hold for all observed data, but this can lead to an overly relaxed PC that the quantizer Q cannot ensure safety (See Remark 1). In practice, we may preprocess the data to remove outliers. For example, the pair of red solid lines in Figure 4b depicts the bounds inferred from ignoring the worst 1% of perceived values and therefore covering 99% of data, and the pair of red dashed lines depicts the bounds covering 90% of data. We also plot the bounds transformed back to the linear scale in Figure 4a. Similarly, selecting larger or smaller θ_b defines more strict or relaxed *constant bounds* for the angular coordinates. It is worth noting that, due to the normalization, angles are wrapped around 0 and 2π as shown in Figure 4c. Therefore, we follow the standard approach to normalize the angle error $\hat{\theta}_i - \theta_i$ to $[-\pi, \pi)$ and infer the bounds as shown in Figure 4d.

The perception contract also depends on environmental factors. To systematically study the impact of environmental variations on the perception contract, we experimented with

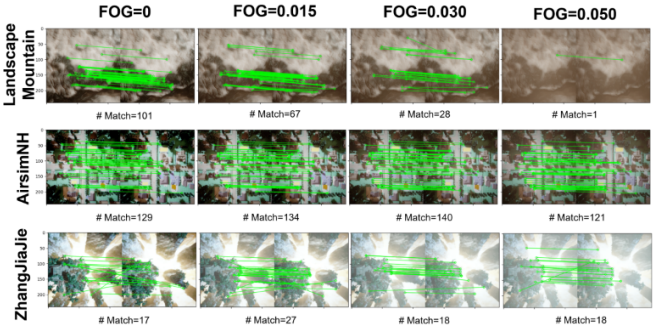


Fig. 5: Matched feature points found by the feature detection and matching algorithm under three AirSim environments and four fog levels.

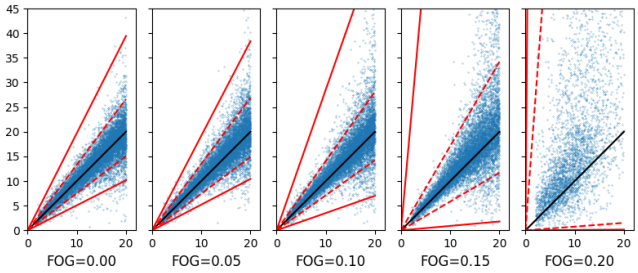


Fig. 6: Perceived distances $\hat{r}_i$ and empirical linear bounds with respect to true distances r_i under five fog levels.

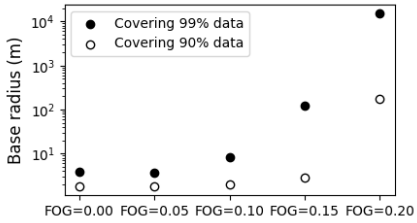


Fig. 7: Inferred step radius a on a log scale covering 99% and 90% of data with respect to five fog levels.

different environments and weather conditions in the photo-realistic AirSim simulator. Figure 5 shows how the feature matching step degrades across three environments (namely LandscapeMountains, AirSimNH, and ZhangJiajie) and four fog levels. Note that at the fog level 0.050, only one pair of matching features is detected for the same relative position under LandscapeMountains.

Figure 6 shows the perception contracts for five fog levels under AirSimNH. The perception bound increases much faster (against the relative distance) in a foggier weather. To better visualize this trend, we further plot the inferred step radius a for covering 99% and 90% of data points under varying fog levels in Figure 7. Note that the y-axis is on a log scale in Figure 7, so both values in fact increase faster than exponential growth with respect to the fog levels, and the step radius for covering 99% data increases more significantly. Unsurprisingly, when it is too foggy, the value of a is too large to satisfy the condition for Theorem 2 to ensure safety.

V. Limitations and Discussions

We presented an analysis for the convergence and safety of a vision-based formation control system. To tackle the vagaries of the perception component, our approach uses a perception contract represented as a quantizer. This quantizer captures the worst perception error in relative position estimates from the vision component, which is then used to prove that the drones are safely separated and converges to the desired formation. Especially, we designed non-uniform quantizers to model the *state-dependent perception error*. We empirically showed that a quantizer in log-polar coordinates models the perception error more accurately using the high-fidelity simulator, AirSim. We systematically studied the impact of environmental variations on the perception contract. We inferred quantization step sizes according to data sampled under each environment so that the instantiated quantizer better models the observed error under the environment.

Our study assumed that all drones run synchronously and exchange image feature descriptors instantly. This is obviously an idealization. Our analysis will work without this assumption by bounding the change in relative positions under a fixed communication delay. We can model the change in relative positions as part of the perception error.

Finally, this paper suggests a broad research direction on connecting quantized control and discrete abstractions over the continuous state space [30]. Both quantization and discrete abstractions are partitioning the state space, but they are different in that operators for quantized values such as difference, averaging, maximum, and minimum are not necessarily available for discrete abstractions. Relating discrete abstractions with quantization will allow us to reuse the theories in quantized control for formal safety analyses.

References

[1] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, "Convergence in multiagent coordination consensus and flocking," in *Proc. Joint 44th IEEE Conf. Decision and Control and Eur. Control Conf.*, 2005, pp. 2996–3000.

[2] R. Saber and R. Murray, "Flocking with obstacle avoidance: cooperation with limited communication in mobile networks," in *Proc. 42nd IEEE Int. Conf. Decision and Control*, vol. 2, 2003, pp. 2022–2028.

[3] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton University Press, 2010.

[4] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton University Press, 2009.

[5] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagüés, "Vision-Based Distributed Formation Control Without an External Positioning System," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 339–351, 2016.

[6] K. Fathian, "Distributed Formation Control of Autonomous Vehicles via Vision-Based Motion Estimation," Ph.D. dissertation, Dept. Elect. Eng., Univ. Texas at Dallas, Richardson, TX, USA, 2018.

[7] M. M. H. Fallah, F. Janabi-Sharifi, S. Sajjadi, and M. Mehrandezh, "A Visual Predictive Control Framework for Robust and Constrained Multi-Agent Formation Control," *J. Intell. Robot. Syst.*, vol. 105, no. 4, 2022.

[8] M. Abraham, A. Mayne, T. Perez, I. R. De Oliveira, H. Yu, C. Hsieh, Y. Li, D. Sun, and S. Mitra, "Industry-track: Challenges in rebooting autonomy with deep learned perception," in *Proc. 2022 Int. Conf. Embedded Softw.*, 2022, pp. 17–20.

[9] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[10] C. Hsieh, Y. Li, D. Sun, K. Joshi, S. Misailovic, and S. Mitra, "Verifying Controllers With Vision-Based Perception Using Safe Approximate Abstractions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4205–4216, 2022.

[11] A. Astorga, C. Hsieh, P. Madhusudan, and S. Mitra, "Perception Contracts for Safety of ML-Enabled Systems," *Proc. ACM on Programming Languages*, vol. 7, no. OOPSLA2, pp. 299:1–299:27, 2023.

[12] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Proc. 11th Int. Conf. Field and Service Robot.*, 2018, pp. 621–635.

[13] C. S. Păsăreanu, R. Mangal, D. Gopinath, S. Getir Yaman, C. Imrie, R. Calinescu, and H. Yu, "Closed-loop analysis of vision-based autonomous systems: A case study," in *Proc. 35th Int. Conf. Comput. Aided Verification*, 2023, pp. 289–303.

[14] R. Calinescu, C. Imrie, R. Mangal, G. N. Rodrigues, C. Păsăreanu, M. A. Santana, and G. Vázquez, "Discrete-event controller synthesis for autonomous systems with deep-learning perception components," arXiv:2202.03360, 2023.

[15] C. Păsăreanu, R. Mangal, D. Gopinath, and H. Yu, "Assumption Generation for the Verification of Learning-Enabled Autonomous Systems," arXiv:2305.18372, 2023.

[16] D. Sun, B. C. Yang, and S. Mitra, "Learning-based perception contracts and applications," 2023.

[17] S. M. Katz, A. L. Corso, C. A. Strong, and M. J. Kochenderfer, "Verification of image-based neural network controllers using generative models," *J. Aerosp. Inf. Syst.*, vol. 19, no. 9, pp. 574–584, 2022.

[18] U. Santa Cruz and Y. Shoukry, "NNLander-VeriF: A Neural Network Formal Verification Framework for Vision-Based Autonomous Aircraft Landing," in *Proc. 14th Int. Symp. NASA Formal Methods*, 2022, pp. 213–230.

[19] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, and S. A. Seshia, "VerifAI: A Toolkit for the Formal Design and Analysis of Artificial Intelligence-Based Systems," in *Proc. 31st Int. Conf. Comput. Aided Verification*, 2019, pp. 432–442.

[20] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. 9th Eur. Conf. Comput. Vision*, 2006, pp. 404–417.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. 2011 Int. Conf. Comput. Vision*, 2011, pp. 2564–2571.

[23] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *Proc. 4th Int. Conf. Comput. Vision Theory Appl.*, 2009, pp. 331–340.

[24] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, 1st ed., ser. Interdisciplinary Applied Mathematics. SpringerVerlag, 2003.

[25] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," INRIA, Research Report RR-6303, 2007. [Online]. Available: https://hal.inria.fr/inria-00174036

[26] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, 2004.

[27] H. Li and R. Hartley, "Five-Point Motion Estimation Made Easy," in *18th Int. Conf. Pattern Recognit.*, vol. 1, 2006, pp. 630–633.

[28] D. Liberzon, "Hybrid feedback stabilization of systems with quantized signals," *Automatica*, vol. 39, no. 9, pp. 1543–1554, 2003.

[29] M. Moakher, "Means and Averaging in the Group of Rotations," *SIAM J. Matrix Anal. Appl.*, vol. 24, no. 1, pp. 1–16, 2002.

[30] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete Abstractions of Hybrid Systems," *Proc. IEEE*, vol. 88, no. 7, pp. 971–984, 2000.