How Fast is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid

Davide Falanga, Suseong Kim, and Davide Scaramuzza

***Abstract*—In this work, we study the effects that perception latency has on the maximum speed a robot can reach to safely navigate through an unknown cluttered environment. We provide a general analysis that can serve as a baseline for future quantitative reasoning for design trade-offs in autonomous robot navigation. We consider the case where the robot is modeled as a linear second-order system with bounded input and navigates through static obstacles. Also, we focus on a scenario where the robot wants to reach a target destination in as little time as possible, and therefore cannot change its longitudinal velocity to avoid obstacles. We show how the maximum latency that the robot can tolerate to guarantee safety is related to the desired speed, the range of its sensing pipeline, and the actuation limitations of the platform (i.e., the maximum acceleration it can produce). As a particular case study, we compare monocular and stereo frame-based cameras against novel, low-latency sensors, such as event cameras, in the case of quadrotor flight. To validate our analysis, we conduct experiments on a quadrotor platform equipped with an event camera to detect and avoid obstacles thrown towards the robot. To the best of our knowledge, this is the first theoretical work in which perception and actuation limitations are jointly considered to study the performance of a robotic platform in high-speed navigation.**

***Index Terms*—Collision Avoidance; Visual-Based Navigation; Aerial Systems: Perception and Autonomy.**

SUPPLEMENTARY MATERIAL

All the videos of the experiments are available at:
`http://youtu.be/sbJAi6SXOQw`

I. INTRODUCTION

HIGH-speed robot navigation in cluttered, unknown environments is currently an active research area [1]–[7] and benefits of over 50 million US dollar funding available through the DARPA Fast Lightweight Autonomy Program (2015-2018) and the DARPA Subterranean Challenge (2018-2021).

To prevent a collision with an obstacle or an incoming object, a robot needs to detect them as fast as possible and execute a safe maneuver to avoid them. The higher the relative speed between the robot and the object, the more critical the role of *perception latency* becomes.

Manuscript received: September, 10, 2018; Revised November, 13, 2018; Accepted January, 30, 2019. This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the SNSF-ERC Starting Grant and the Swiss National Science Foundation through the National Center of Competence in Research (NCCR) Robotics.

The authors are with the Robotics and Perception Group, Dep. of Informatics, University of Zurich, and Dep. of Neuroinformatics, University of Zurich and ETH Zurich, 8050 Zurich, Switzerland.

Digital Object Identifier (DOI): see top of this page.

Perception latency is the time necessary to *perceive* the environment and *process* the captured data to generate control commands. Depending on the task, the processing algorithm, the available computing power, and the sensor (e.g., lidar, camera, event camera, RGB-D camera), the perception latency can vary from tens up to hundreds of milli-seconds [2]–[7].

At the current state of the art, the agility of autonomous robots is bounded, among the other factors (such as their actuation limitations), by their sensing pipeline. This is because the relatively high latency and low sampling frequency limit the aggressiveness of the control strategies that can be implemented. It is typical in current robots to have latencies of tens or hundreds of milli-seconds. Faster sensing pipelines can lead to more agile robots.

Despite the importance of the perception latency, very little attention has been devoted to study its impact on the agility of a robot for a sense and avoid task. Analyzing the role of sensing latency allows one to understand the limitations of current perception systems, as well as to comprehend the benefits of exploiting novel image sensors and processors, such parallel visual processors (e.g., SCAMP [8]), with a theoretical latency of few milli-seconds, or event cameras, with a theoretical latency of micro-seconds (e.g., the DVS [9]) or even nano-seconds (e.g., CeleX [10]).

In the context of robot navigation, it is also important to correlate the sensing latency to the actuation capabilities of the robot. Broadly speaking, the larger the acceleration a robot can produce, the lower the time it needs to avoid an obstacle and, therefore, the larger the latency it can tolerate. Consequently, the coupling between sensing latency and the actuation limitations of a robot represents a key research problem to be addressed.

A. Related Work

Sensing latency is a known issue in robotics and has already been investigated before. For example, this problem is particularly interesting when the state estimation process is done through visual localization. A number of vision-based solutions for low-latency localization based either on standard cameras [11], [12] or novel sensors (e.g., event cameras [2], [13], [14]) have been proposed. Impressive results have been achieved, however no information about the environment is available since visual localization only provides the robot the information about its pose.

It is not yet clear what the maximum latency of a perception system for a navigation task should be. A first step in that direction is available in [15], where the authors studied under which circumstances a high frame-rate is best for real-time

tracking, providing quantitative results that help selecting the optimal frame-rate depending on required performance. The results of that work were tailored towards visual localization for state estimation. In [16] the performance of visual servoing as a function of a number of parameters describing the perception system (e.g., frame-rate, latency) was studied, and a relation between the tracking error in the image plane and the latency of the perception was derived.

In [17], a framework to predict and compensate for the latency between sensing and actuation in a robotic platform aimed at visually tracking a fast-moving object was proposed and experimental results showed the benefits of that framework. Nevertheless, the impact of the latency on the performance of the executed task without the proposed compensation framework was not discussed.

The most similar work to ours is [18], where the authors studied the performance of vision-based navigation for mobile robots depending on the latency and the sensing range of the perception system. A trade-off among camera frame rate, resolution, and latency was shown to represent the best configuration for navigation in unstructured terrain. However, such results were only supported by experimental results, without any theoretical evidence. Different from our work, the actuation capabilities of the robot were not considered.

To the best of our knowledge, no previous works analyzed the coupling between sensing latency and actuation limitations in a robotic platform from a theoretical perspective. Similarly, the problem of highlighting their impact on the performance of high-speed navigation has not been addressed in the literature.

B. Contributions

In this work, we focus on the effects of perception latency and actuation limitations on the maximum speed a robot can reach to safely navigate through an unknown, static scenario.

We consider the case where a generic robot, modeled as a linear system with bounded inputs, moves in a plane and relies on onboard perception to detect static obstacles along its path (cf. Fig. 1). We focus on a scenario where the robot wants to reach a target destination in as little time as possible, and therefore cannot change its longitudinal velocity to avoid obstacles. We show how the maximum latency the robot can tolerate to guarantee safety is related to the desired speed, the agility of the platform (e.g., the maximum acceleration it can produce), as well as other perception parameters (e.g., the sensing range). Additionally, we derive a closed-form expression for the maximum speed that the robot can reach as a function of its perception and actuation parameters, and study its sensitivity to such parameters.

We provide a general analysis that can serve as a baseline for future quantitative reasoning for design trade-offs in autonomous robot navigation, and is completely agnostic to the sensor and robot type. As a particular case study, we compare standard cameras against event cameras for autonomous quadrotor flight, in order to highlight the potential benefits of these novel sensors for perception. Finally, we provide an experimental evaluation and validation of the proposed theoretical analysis for the case of a quadrotor, equipped with an event camera, avoiding a ball thrown towards it at speeds up to $9\,\mathrm{m/s}$.

To the best of our knowledge, this is the first work in which perception and actuation limitations are jointly considered to study the performance of a robot in high-speed navigation.

C. Assumptions

This works is based on the following assumptions. First, we assume that the robot can be model as a linear system. Robotic systems are typically characterized by non-linear models. However, a large variety of them can be linearized through either static or dynamic feedback [19], rendering them equivalent from a control perspective to a chain of integrators. It is important to note that feedback linearization is different from Jacobian linearization: the first is an exact representation of the original non-linear system over a large variety of working conditions, while the second is only valid locally [20]. Linear models for mobile robots have already been used in the past [1], and come with the advantage of allowing a simple, yet effective mathematical analysis of the behaviour of the system in closed-form. Also, they cover a large variety of systems, rendering our analysis valid for different kinds of robots.

Second, we assume that the robot can execute holonomic 2D maneuvers. For non-holonomic systems, such as fixed-wing aircraft, the coupling of the longitudinal and lateral dynamics would break the assumptions of our model and would deserve a different analysis.

Finally, since we are interested in the role of sensing latency and actuation limitations on the agility of a robot, we assume that, for any other aspect, the sensing and actuation system are ideal. In other words, we assume that there is no uncertainty in the obstacle detection, no illumination issues, no artifacts in the measurements, and the robot's dynamics is perfectly known and can be controlled with errors. This allows us to clearly isolate and analyze the impact of sensing latency and actuation limitations in our analysis, where otherwise it would not be possible to distinguish the role of these two from the impact of other sources of non-ideality.

D. Structure of the Paper

In Sec. II, we provide the mathematical formulation of the problem and perform a qualitative analysis. In Sec. III, we particularize our study to vision-based navigation and analyze it for both standard and event cameras. A detailed mathematical analysis of these sensors is provided in the supplementary material. In Sec. IV, we compare standard cameras (monocular and stereo) against event cameras for the case study of autonomous quadrotor flight. In Sec. V, we validate our analysis performing experiments on an actual quadrotor avoiding obstacle thrown towards it. Further details about the experiments are provided in the supplementary material. Finally, in Sec. VI, we draw the conclusions.

II. Problem Formulation

We consider the case of a mobile robot navigating in a plane, which covers a large number of scenarios, e.g. an aerial robot

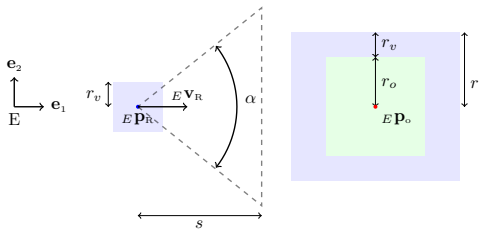

Fig. 1: A schematics representing the obstacle and the robot model in the frame E. The robot is represented as a square of size $2r_v$ centered at ${}_E\mathbf{p}_\mathrm{R}$, and moves with a speed ${}_E\mathbf{v}_\mathrm{R}$. The dashed triangle starting from the robot's position represents its sensing area, α is the field of view and s the maximum distance it is able to perceive. The obstacle, represented by the green square on the right side of the image, has size $2r_o$. We expand the square representing the obstacle by a quantity r_v such that the robot can be considered to be a point mass.

flying in a forest [1], where the third dimension would not help with the avoidance task. The robot moves along a desired direction with a desired speed, provided by a high-level planner, towards its goal, which has to be reached in as little time as possible. Therefore, the robot cannot change its longitudinal velocity. In the following analysis, we consider the case where the robot only faces one single obstacle along its path and then provide an intuitive explanation of how our conclusions can be extended to the case of multiple obstacles.

A. Modelling

1) Robot Model: Let E be the inertial reference frame, having basis $\{\mathbf{e}_1, \mathbf{e}_2\}$, and let ${}_E\mathbf{p}_\mathrm{R}$ and ${}_E\mathbf{v}_\mathrm{R}$ be the position and velocity, respectively, of the robot in E. Also, let ${}_E\mathbf{p}_\mathrm{O}$ be the position of an obstacle in E. In the remainder, we will refer to $\mathbf{e}_1$ as the *longitudinal* axis, and $\mathbf{e}_2$ as the *lateral* axis. Finally, let r_v be the half-size of the square centered at ${}_E\mathbf{p}_\mathrm{R}$ containing the entire robot (cf. Fig. 1).

We model both the longitudinal and lateral dynamics as a chain of integrators. As shown in [19], a large variety of mechanical systems can be linearized by using nonlinear feedback, which, from a control perspective, renders them equivalent to a chain of integrators. Additionally, the dynamics of the actuators is usually faster than the mechanical dynamics and can, therefore, be neglected.

The longitudinal and lateral dynamics are modeled by a position $\mathrm{p_i}$, a speed $\mathrm{v_i}$ and an input $\mathrm{u_i}$ given by:

$$\dot{\mathrm{p}}_1(t) = \mathrm{v}_1(t), \quad \dot{\mathrm{v}}_1(t) = \mathrm{u}_1(t), \tag{1}$$

$$\dot{\mathrm{p}}_2(t) = \mathrm{v}_2(t), \quad \dot{\mathrm{v}}_2(t) = \mathrm{u}_2(t). \tag{2}$$

Both inputs are assumed to be bounded such that $\mathrm{u_i} \in [-\bar{\mathrm{u}}_\mathrm{i}, \bar{\mathrm{u}}_\mathrm{i}], i = 1, 2$. We assume the robot to move only along the longitudinal axis with an initial speed $\mathrm{v}_{1,0} = \hat{\mathrm{v}}_1$, meaning that the lateral speed is zero before the avoidance maneuver starts. The case where the robot has non-zero lateral velocity can be analyzed using the same mathematical framework. Also, we assume that the robot cannot change its longitudinal speed, namely $\mathrm{u}_1(t) = 0\ \forall t$, and can therefore only exploit the lateral dynamics to avoid an obstacle. As shown in Sec. S1 of the supplementary material, a lateral avoidance maneuver requires less time at high speed, allowing faster navigation along the longitudinal axis.

2) Obstacle Model: We consider static obstacles enveloped by a square of width $2r_o$. To study the motion of the robot considering only the position of its center, we expand the obstacle width by a quantity r_v on each side. The expanded size of the obstacle is $r = 2(r_v + r_o)$, as shown in Fig. 1.

3) Sensor Model: In this work, we assume that at least one edge of the obstacle must enter the sensing area to allow a detection. We define the sensing latency $\tau \in \mathbb{R}^+$ as the interval between the time the obstacle enters the sensing area and the moment the robot's initiates the avoidance maneuver. The latency of a sensor is typically the sum of multiple contributions, and in general depends on the sensor itself and the time necessary to process a measurement (which depends on the algorithm used, the computational power available, and other factors). In general, it is hard to provide exact bounds for each of these contributions, therefore we consider as latency the sum of the sensor's and the sensing algorithm's latency. We denote by $s \in \mathbb{R}^+$ the robot's sensing range, i.e. the largest distance it is able to perceive. We assume the field of view of the sensor to be such that the obstacle's edge is fully contained in the sensing area when the distance between the robot and the obstacle is equal to the sensing range. This provides a lowerbound for the field of view $\alpha \geq 2 \arctan\left(\frac{r_o}{2s}\right)$.

B. Obstacle Avoidance

1) Time to Contact and Avoidance Time: We define the *time to contact* t_c as the time it takes the vehicle to collide with the obstacle once it enters the sensing range of its onboard sensor. Since the longitudinal motion has a constant speed $\hat{\mathrm{v}}_1$ and the distance between the vehicle and the obstacle at the time the obstacle enters the sensing area is s, the time to contact t_c is:

$$t_\mathrm{c} = \frac{s}{\hat{\mathrm{v}}_1}. \tag{3}$$

In order for the robot to avoid the obstacle, it has to reach a safe lateral position in an *avoidance time* t_s shorter than the time to contact (3).

$$t_\mathrm{c} \geq t_\mathrm{s}. \tag{4}$$

2) Time-Optimal Avoidance: The avoidance maneuver along the lateral axis leads to a safe navigation if $\mathrm{p}_2(t_\mathrm{c}) \geq r$. We consider the case $\mathrm{p}_2(t_\mathrm{c}) = r$, which represents the minimum lateral deviation for the avoidance maneuver to be executed safely. For this to happen, we assume the robot to use a time-optimal strategy $\mathrm{u}_2^*(t)$:

$$\begin{aligned} \mathrm{u}_2^*(t) = \underset{\mathrm{u}_2(t)}{\arg\min} \quad & t_\mathrm{s} \\ \text{subject to} \quad & \dot{\mathrm{p}}_2(t) = \mathrm{v}_2(t),\ \dot{\mathrm{v}}_2(t) = \mathrm{u}_2(t), \\ & \mathrm{p}_2(0) = 0, \quad \mathrm{v}_2(0) = 0, \\ & \mathrm{p}_2(t_\mathrm{s}) = r, \quad \mathrm{v}_2(t_\mathrm{s}) = 0, \\ & \mathrm{u}_2(t) \in [-\bar{\mathrm{u}}_2, \bar{\mathrm{u}}_2]\ \forall t. \end{aligned} \tag{5}$$

We require $\mathrm{v}_2(t_\mathrm{s}) = 0$ because there would be no advantage in having a non-zero lateral speed in terms of progressing towards

the goal, since we considered the longitudinal axis to be the direction of motion. Leaving the final lateral speed free would lead to a lower execution time for the avoidance maneuver, but this could potentially result in a large lateral speed, which is typically not desirable because the robot is not able to sense the environment in such a direction. As well known in the literature [21], the problem (5) leads to a *bang-bang* solution:

$$\mathrm{u}_2^*(t) = \begin{cases} \bar{\mathrm{u}}_2 & \text{if } 0 \leq t \leq \hat{t} \\ -\bar{\mathrm{u}}_2 & \text{if } \hat{t} < t \leq t_\mathrm{s} \end{cases}, \tag{6}$$

where the $\hat{t} = \sqrt{\frac{r}{\bar{\mathrm{u}}_2}}$ is the switching time and $t_\mathrm{s} = 2\sqrt{\frac{r}{\bar{\mathrm{u}}_2}}$ is the avoidance time..

3) Obstacle Avoidance with Sensing Latency: In Sec. II-B1 we defined the time to contact t_c as the time between when the obstacle enters the sensing range and the moment when the collision occurs, as defined in (3). However, in the presence of sensing latency, the time $t_\mathrm{c}^{'}$ remaining to the collision when the robot is informed about the presence of the obstacle is $t_\mathrm{c}^{'}(\tau) = t_\mathrm{c} - \tau$. Therefore, in order for a robot equipped with a sensor with sensing range s and latency τ to safely avoid an obstacle, the condition $t_\mathrm{c}^{'}(\tau) \geq t_\mathrm{s}$ must hold. In this case, we can compute (4) as:

$$\frac{s}{\hat{\mathrm{v}}_1} - \tau \geq 2\sqrt{\frac{r}{\bar{\mathrm{u}}_2}}. \tag{7}$$

The worst case in which the robot manages to avoid the obstacle occurs when (7) is satisfied with equality. In this case, the robot passes tangent to the obstacle, whereas it would have some safety margin if (7) was satisfied with the inequality sign. We can study (7) to compute the maximum latency $\bar{\tau}$ the system can tolerate such that the avoidance can still be performed safely:

$$\bar{\tau} = \frac{s}{\hat{\mathrm{v}}_1} - 2\sqrt{\frac{r}{\bar{\mathrm{u}}_2}}. \tag{8}$$

Fig. 2 shows the maximum latency $\bar{\tau}$ for different values of $\bar{\mathrm{u}}_2$ and s for the case $r = 0.5\,\mathrm{m}$. As one can notice, the importance of low latency increases as the navigation speed increases. Also, for some speeds $\hat{\mathrm{v}}_1$ the robot is unable to perform the avoidance maneuver safely given its actuation capabilities and the sensing range of its sensor. This is clear from the negative values the maximum latency $\bar{\tau}$ assumes in some intervals. In this case the robot should be either more agile (i.e. capable of generating higher lateral accelerations) or should be equipped with a sensor with a higher sensing range in order to avoid the obstacle at such speeds.

Similarly, we can use (8) to compute the maximum longitudinal speed the robot can have to avoid the obstacle:

$$\bar{\mathrm{v}}_1 = \frac{s}{\tau + 2\sqrt{\frac{r}{\bar{\mathrm{u}}_2}}}. \tag{9}$$

Fig. 3 shows the maximum speed the robot can navigate safely (i.e., being still able to avoid the obstacle although this is perceived with some delay), depending on the latency of its sensing pipeline.

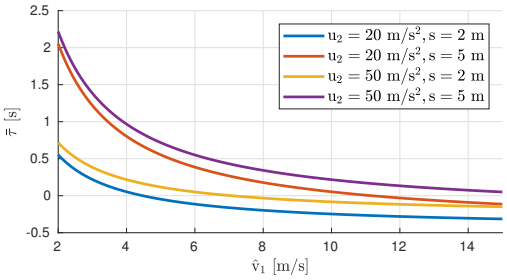

Fig. 2: Maximum latency $\bar{\tau}$ that the robot can tolerate in order to safely perform the avoidance maneuver when $r = 0.5\,\mathrm{m}$.

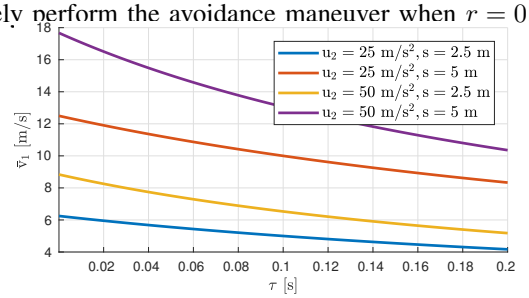

Fig. 3: Maximum speed $\bar{\mathrm{v}}_1$ that the robot can move in order to safely perform the avoidance maneuver when $r = 0.5\,\mathrm{m}$.

III. Vision-Based Perception

In the following, we particularize our analysis to the case of vision-based perception for three modalities: (i) a monocular frame-based camera, (ii) a stereo frame-based camera, (iii) a monocular event camera, and analyze the impact of their latency on the maximum speed. For brevity reasons, the mathematical derivation of the expressions for the sensing range and the latency of each of these sensing modalities is reported in the supplementary material attached to this work.

A. Frame-Based Cameras and Event Cameras

Most computer vision research has been devoted to frame-based cameras, which have latencies in the order of tens of milli-seconds, thus, putting a hard bound on the achievable agility of a robotic platform. By contrast, event cameras [9] are bio-inspired vision sensors that output pixel-level brightness changes at the time they occur, with a theoretical latency of micro-seconds or even nano-seconds. More specifically, rather than streaming frames at constant time intervals, each pixel *fires* an event (a pixel-level brightness change), independently of the other pixels, every time it detects a change of brightness in the scene. Broadly speaking, we can consider event cameras as *motion-activated*, asynchronous edge detectors: events fire only if there is relative motion between the camera and the scene.

Exploiting frame-based cameras for obstacle avoidance typically requires the analysis of all the pixels of the image to detect an obstacle, independently of the texture. Conversely, since the pixels of an event camera only trigger information when there is change of intensity, it has the advantage of requiring very little processing to detect an obstacle. Furthermore, since the smallest time interval between two consecutive events on the same pixel is in the order of $1\,\mu\mathrm{s}$, or generally much smaller than the typical framerate of frame-based cameras, this

can safely be neglected. These factors result in a theoretical advantage of event cameras against frame-based cameras.

B. Sensing Range of a Vision-Based Perception System

1) Monocular Frame-Based Camera: The sensing range s_{M} of a monocular camera depends, as shown in Sec. S4-A of the supplementary material, on the size r_o of the obstacle, the number of pixels N it must occupy in the image to be detected, and the camera's angular resolution θ.

2) Stereo Frame-Based Camera: The sensing range s_{S} of a stereo camera depends, as shown in Sec. S5-A of the supplementary material, on the baseline b, the focal length f, the uncertainty in the disparity ϵ_{P} and the maximum percentual uncertainty k in the depth estimation.

3) Event Camera: In Sec. S6-A of the supplementary material we show that the sensing range s_{E} of an event camera can be computed using (S.1). It depends on how large the object must be in the image such that, when its edges generate an event, they are sufficiently far apart.

C. Latency of a Vision-Based Perception System

1) Monocular Frame-Based Camera: The latency τ_{M} of a monocular camera depends on the time t_{f} between two consecutive triggers of the sensor, the exposure time t_{E}, the transfer time t_{T}, the processing time and the number of images necessary to detect the obstacle. As shown in Sec S4-B of the supplementary material, if two consecutive images are sufficient to detect an obstacle, it can vary between $\tau_{\mathrm{M}} = t_{\mathrm{f}} + t_{\mathrm{T}} + t_{\mathrm{E}}$ and $\tau_{\mathrm{M}} = 2t_{\mathrm{f}}$.

2) Stereo Frame-Based Camera: In Sec. S5-B of the supplementary material, we analyze the possible range of the latency τ_{S} of a stereo camera. In general, it can span between a best-case value equal to the time between two consecutive frames, and a worst-case value, which we derive analyzing the datasheet of several stereo cameras.

3) Event Camera: The latency τ_{E} of an event camera depends, as shown in Sec. S6-B of the supplementary material, on the distance between the camera and the obstacle, the speed of the camera, the focal length, and the amount of pixels the projection of the obstacle must move in the image such that it fires an event. However, to derive the maximum speed achievable with an event camera, it is necessary to jointly consider the expression of the latency of an event camera and (4). We refer the reader to Sec. S6-B of the supplementary material for further details.

IV. Case Study: Vision-Based Quadrotor Flight

In this section, we analyze the case of vision-based quadrotor flight. We consider a quadrotor equipped with a sensing pipeline based on frame-based cameras in a monocular and stereo configuration, and a monocular event camera. For each sensing modality, we provide an upper and a lower-bound of the sensing range and the latency according to the model in Sec. III. We compute the maximum speed achievable with each sensor for a value of each parameter equal to its lower-bound, its upper-bound, and the average between the upper and the lower-bound. Finally, we consider four different values for the maximum lateral acceleration the quadrotor can produce. Three values correspond to commercially available state-of-the-art quadrotors with low, medium and high *thrust-to-weight* ratio. The fourth one, instead, represents a quadrotor with a *thrust-to-weight* ratio that is, as of today, particularly hard to achieve with current technology, but might become common in the future. This ideal platform serves us to show that more agile quadrotors would significantly highlight the benefits of lower-latency sensors for obstacle avoidance.

A. Sensing Range

1) Monocular Frame-Based Camera: We use the results of Sec. S4-A of the supplementary material to obtain the upper-bound and the lower-bound for the sensing range of a monocular camera. The best-case scenario occurs when the obstacle to be detected occupies 5% of the image, leading to an upper-bound $s_{\mathrm{M}} = 6\,\mathrm{m}$. We consider as worst-case scenario when the obstacle occupies 10%, leading to a lower-bound $s_{\mathrm{M}} = 2\,\mathrm{m}$.

2) Stereo Frame-Based Camera: We assume the robot to be equipped with a stereo system having a baseline $b = 0.10\,\mathrm{m}$ and each camera having a VGA resolution. As shown in Sec. S5-A of the supplementary material, we consider $s_{\mathrm{S}} = 2\,\mathrm{m}$ and $s_{\mathrm{S}} = 8\,\mathrm{m}$ to be reasonable values for the lower-bound and the upper-bound of the sensing range.

3) Event Camera: As mentioned in Sec. S6-A of the supplementary material, the sensing range of an event camera can reach values above $s_{\mathrm{E}} = 10\,\mathrm{m}$. Intuitively speaking, this is because to potentially detect an obstacle with an event camera, it is sufficient that the projection of its edges move on the image by $1\,\mathrm{pxl}$ and are far apart from each others by an amount that is at least on order of magnitude larger (i.e., at least $10\,\mathrm{pxl}$ apart). However, to render our comparison more fair and realistic, we consider a lower-bound that is comparable to the one of frame cameras. Indeed, when a robot navigates cluttered environments, its distance from the obstacles is typically lower than $10\,\mathrm{m}$, which makes it necessary to consider a lower value for the smallest sensing range of event camera. Therefore, we assume $s_{\mathrm{E}} = 2\,\mathrm{m}$ as lower-bound for the sensing range of an event camera, and $s_{\mathrm{E}} = 8\,\mathrm{m}$ as its upper-bound.

B. Latency

1) Monocular Frame-Based Camera: We consider a frame-based camera with (i) a framerate of $50\,\mathrm{Hz}$, meaning that $t_{\mathrm{f}} = 0.020\,\mathrm{s}$; (ii) an exposure time of $t_{\mathrm{E}} = 0.005\,\mathrm{s}$; (iii) VGA resolution and USB 3.0 connection, which leads to $t_{\mathrm{T}} = 0.0004\,\mathrm{s}$. Therefore, based on Sec. S4-B of the supplementary material, the upper-bound and the lower-bound latency for the frame-based camera considered in this analysis are, respectively, $\tau_{\mathrm{M}} = 0.040\,\mathrm{s}$ and $\tau_{\mathrm{M}} = 0.026\,\mathrm{s}$.

2) Stereo Frame-Based Camera: As mentioned in Sec. S5-B of the supplementary material, it is hard to evaluate the latency of a stereo system. However, based on the datasheet of commercially available stereo cameras suitable for quadrotor flight, we can obtain an estimate of the upper-bound and the lower-bound. As upper-bound, we consider the Bumblebee

XB3, whose datasheet reports a latency of $\tau_S = 0.070\,\mathrm{s}$. For the lower-bound, since no further information are available in the datasheet of other stereo cameras, we assume it to be equal to the inverse of the frame-rate of the fastest available sensor (Intel RealSense R200) leading to $\tau_S = 0.017\,\mathrm{s}$.

3) Event Camera: In Sec. S6-B of the supplementary material we discuss how the latency of an event camera depends on the relative distance and speed between the robot and the obstacle. Also, we highlight that, in order to compute it, it is necessary to jointly consider the sensing range (Sec. S6-A), Eq. (8) and Eq. (S.5). Therefore, to analyze the maximum speed achievable with an event camera we proceed as follows: (i) we consider a value of the sensing range as described in Sec. III-B3; (ii) we plug (9) into (S.5) and solve it for $\hat{\mathrm{v}}_1$ to compute the maximum speed achievable; (iii) we use (8) to obtain the corresponding value of the latency of an event camera, given its distance from the obstacle and its speed.

C. Quadrotor Model

The dynamical model of a quadrotor is differentially flat and the vehicle can be considered as a linear system using nonlinear feedback linearization [22] both from a control [23] and a planning perspective [24]. We considered four cases for the maximum lateral acceleration the robot can produce: $\bar{\mathrm{u}}_2 = 10\,\mathrm{m/s^2}$, $\bar{\mathrm{u}}_2 = 25\,\mathrm{m/s^2}$, $\bar{\mathrm{u}}_2 = 50\,\mathrm{m/s^2}$, and $\bar{\mathrm{u}}_2 = 200\,\mathrm{m/s^2}$. These values correspond to a *thrust-to-weight ratio* of approximately 1.5, 2.8 5.2 and 20, respectively. The first three cover a large range of the lift capabilities of commercially available drones, while the fourth represents a vehicle currently not yet available, but which might be available in the future. We assume $r_v = 0.25\,\mathrm{m}$ and $r_o = 0.50\,\mathrm{m}$, leading to an expanded obstacle size of $r = 0.75\,\mathrm{m}$.

D. Results

The results of our analysis for vision-based quadrotor flight are available in Table I. For each sensing modality (first column) we combined three values for the sensing range (second column) and the latency (third column), and computed the maximum speed the robot can achieve depending on the maximum lateral acceleration it can produce (fourth column). For frame-based camera (monocular and stereo), we considered as values for the sensing range and the latency the lower-bound, the upper-bound and the average between upper-bound and lower-bound.

Similarly, we considered three values for the sensing range of an event camera. However, as mentioned in Sec. IV-B3, the latency of event cameras is strictly connected to the robot's agility. As shown in Sec. S6-B of the supplementary material, the theoretical latency of an event camera depends on both its distance to the obstacle and its velocity towards it (c.f. Eq. (S.5)). Broadly speaking, the faster the robot, the earlier the desired amount of events for the detection are generated. However, for the obstacle avoidance problem to be well-posed, the robot cannot be arbitrarily fast, but its speed must be such that the avoidance maneuver requires an amount of time smaller than the time to contact (Eq. (4) and (7)). This means that the theoretical latency of an event camera depends also on the maximum lateral input the robot can produce. Therefore, for a given sensing range and robots maximum input, one can compute the corresponding maximum velocity achievable and, consequently, the latency of an event camera mounted on such a robot. Since different robots maximum input would produce different maximum velocity, the same event camera will similarly have different latencies (Eq. (S.5)). This motivates the dashed values in Table I.

As one can notice, when the sensing range and the robot's agility are small, the difference among mononocular frame cameras, stereo frame cameras and event cameras is not remarkable. Conversely, frame cameras in stereo configuration and event cameras allow faster flight than a monocular frame camera when either the sensing range or the robot's agility increase. In particular, increasing the sensing range, as expected from Sec. S2, allows the robot to navigate faster thanks to a sensible increase of the time to contact.

Similarly, making the robot more agile (i.e., increasing $\bar{\mathrm{u}}_2$) allows it to fly faster thanks to the decrease of the avoidance time. As one can notice by the results in the column of the quadrotor having and $\bar{\mathrm{u}}_2 = 200\,\mathrm{m/s^2}$, the difference between the maximum speed achievable with stereo frame-based cameras and event cameras become significant. Depending on the sensing range, low-latency event cameras allow the robot to reach a maximum speed that can be between 7% and 12% larger than the one achievable with a stereo frame-based camera. It is important to remark that, despite the numbers provided for the case $\bar{\mathrm{u}}_2 = 200\,\mathrm{m/s^2}$ are very high, they are not as far as one could think from what is currently achievable by agile quadrotors. Indeed, First-Person-View (FPV) quadrotors are currently capable of reaching speeds above $40\,\mathrm{m/s}$ with thrust-to-weight ratios above 10 and, given the pace of the technological progress in the FPV community, it is not hard to believe that, in the near future, quadrotors will be able to reach speeds significantly beyond the current values. In FPV racing, a small increase in the maximum flight speed can represent the step necessary to outperform other vehicles participating in the race. This is particularly interesting in the contest of autonomous FVP drone racing, an extremely active area of research [25], [26].

V. Experiments

To validate our analysis, we performed real-world experiments with a quadrotor platform equipped with an Insightness SEEM1 sensor [1], a very compact neuromorphic camera providing standard frame, events and Inertial Measurement Unit data. The obstacle was a ball of radius $10\,\mathrm{cm}$ thrown towards the quadrotor, and the vehicle only relied on the onboard event camera to detect it and avoid it. From the perspective of our model, this is equivalent to the case where the robot moves towards the obstacle, since the time to contact depends on the absolute value of the relative longitudinal velocity. This experimental setup allowed us to reach large relative velocities in a confined space. Further details about the experimental platform used in this work are available in Sec.S8-A of the supplementary material.

[1] http://www.insightness.com/technology

Sensor Type	**Sensing Range** [m]	**Latency** [s]	**Max. speed** [m/s]			
			$\bar{u}_2 = 10\,\mathrm{m/s^2}$	$\bar{u}_2 = 25\,\mathrm{m/s^2}$	$\bar{u}_2 = 50\,\mathrm{m/s^2}$	$\bar{u}_2 = 200\,\mathrm{m/s^2}$
Mono Frame	2.0	0.026	3.48	5.37	7.38	13.47
	2.0	0.033	3.44	5.27	7.20	12.83
	2.0	0.040	3.40	5.17	7.02	12.30
	4.0	0.026	5.23	8.06	11.07	26.94
	4.0	0.033	5.17	7.91	10.79	25.73
	4.0	0.040	5.10	7.76	10.53	24.62
	6.0	0.026	6.97	10.74	14.76	40.41
	6.0	0.033	6.89	10.54	14.39	38.59
	6.0	0.040	6.81	10.35	14.03	36.93
Stereo Frame	2.0	0.017	3.54	5.51	7.64	14.37
	2.0	0.043	3.38	5.13	6.93	12.06
	2.0	0.070	3.24	4.80	6.35	10.39
	5.0	0.017	8.86	13.77	19.11	35.93
	5.0	0.043	8.50	12.83	17.34	30.16
	5.0	0.070	8.10	12.01	15.88	25.98
	8.0	0.017	14.17	22.03	30.57	57.50
	8.0	0.043	13.54	20.53	27.75	48.25
	8.0	0.070	12.95	19.21	25.40	41.56
Mono Event	2.0	0.002	-	-	-	16.12
	2.0	0.003	-	-	8.06	-
	2.0	0.004	-	5.70	-	-
	2.0	0.007	3.60	-	-	-
	5.0	0.004	-	-	-	39.53
	5.0	0.008	-	-	19.76	-
	5.0	0.011	-	13.98	-	-
	5.0	0.017	8.84	-	-	-
	8.0	0.006	-	-	-	62.06
	8.0	0.012	-	-	31.03	-
	8.0	0.018	-	21.94	-	-
	8.0	0.029	13.88	-	-	-

TABLE I: The results of our case study. We compare monocular frame-based cameras, stereo frame-based cameras and event cameras for different robot agility values. The dashes in the columns reporting the maximum speed achievable with an event camera are due to the fact that, given a value for the sensing range and the maximum lateral acceleration, we can compute the maximum achievable speed and the corresponding latency (c.f. Sec. IV-D for a more detailed explanation).

A. Obstacle Detection with an Event Camera

To detect the obstacle, whose size is supposed to be known, we use a variation of the algorithm proposed in [27] to remove events generated by the static part of the environment due to the motion of the camera. Different from [27], we do not compensate for the camera's motion using numerical optimization, but rather exploiting the gyroscope's measurements. This allows our pipeline to be faster, but comes at the cost of a higher amount of not compensated events.

We accumulate motion-compensated events over a sliding window of $10\,\mathrm{ms}$, obtaining an *event-frame* containing the timestamp of the events due to the motion of moving objects. Such event-frame typically consists of several separated blobs, which are clustered together using the DBSCAN algorithm [28] based on their relative distance, their direction of motion (obtained using Lucas-Kanade tracking [29]) and the timestamp of the events. We fit a rectangle around the blobs belonging to the same cluster and look for the rectangle having the most similar aspect ratio to the expected one. Since we assume the size of the obstacle to be known, we compute its expected aspect ratio and, after finding the most similar cluster, we project its the centroid into the world frame using the standard pinhole camera projection model.

To render our algorithm most robust to outliers, we considered the obstacle to be detected only when at least n measurements in the world frame are obtained and their relative distance is below a threshold. Our experimental evaluation showed that 2 consecutive measurements at a relative distance lower than $20\,\mathrm{cm}$ were sufficient to detect the ball in a reliable way. Also, we fixed the sensing range by discarding detections happening when the ball was at a distance from the robot larger than its sensing range.

It is important to note that our detection algorithm was designed with the aim of reducing the latency of the sensing pipeline and, during the tuning stage, speed was prioritized against accuracy. Accurate obstacle detection with event cameras of obstacles of unknown size and shape is beyond of the scope of this paper.

B. Expected and Measured Latency

Theoretically, a $1\,\mathrm{pxl}$ motion of the projection of point in the image is sufficient to generate an event. However, in our experiment we realized that a larger motion is necessary to obtain reliable obstacle detection with an event camera. More specifically, the algorithm was able to detect the obstacle thrown towards the vehicle whenever a displacement between of at least $5\,\mathrm{pxl}$ was verified. In Sec. S8-C of the supplementary material we analyze this aspect and discuss the main reasons causing the discrepancy between the theoretical ideal model and real data. Also, we exploited the model proposed in Sec. S6-B of the supplementary material to compute the theoretical latency for an event camera having the same resolution of the sensor used in our experiments, for a pixel displacement of $5\,\mathrm{pxl}$. Sec. S8-B of the supplementary material reports the theoretical latency for an obstacle detection pipeline based on an Insightness SEEM1, and the measured latency for our algorithm. As one can see from Fig. 9, Fig. 10, and Tab. I in the supplementary material, the experimental data agree with the theoretical model. Sec. S8-C of the supplementary material discusses the discrepancy between our model and actual data.

C. Results

We performed experiments where the quadrotor described in Sec. S8-A of the supplementary material, equipped with an Insightness SEEM1 sensor and running the detection algorithm described in Sec. V-A, was commanded to avoid a ball thrown towards it. The ball was thrown with a speed spanning between $\hat{\mathrm{v}}_1 = 5\,\mathrm{m/s}$ and $\hat{\mathrm{v}}_1 = 9\,\mathrm{m/s}$. The sensing range was $2\,\mathrm{m}$, meaning that any detection at distance larger than this amount was neglected. Therefore, the time to contact spanned between $t_\mathrm{c} = 0.22\,\mathrm{s}$ and $t_\mathrm{c} = 0.40\,\mathrm{s}$. The robot was commanded to execute an avoidance maneuver either upwards, laterally or diagonally. The obstacle radius was $r_o = 10\,\mathrm{cm}$, while the robot's size was computed as either its height ($r_v = 15\,\mathrm{cm}$) or half its tip-to-tip diagonal ($r_v = 25\,\mathrm{cm}$), depending on the direction of the the avoidance maneuver. Therefore, the expanded obstacle radius spanned between $r = 25\,\mathrm{cm}$ and $r = 35\,\mathrm{cm}$. The avoidance spanned between $t_\mathrm{s} = 0.17\,\mathrm{s}$ and $t_\mathrm{s} = 0.25\,\mathrm{s}$. In all the experiments, the ball would have hit the vehicle if the avoidance maneuver was not executed, as confirmed by ground truth data provided by the motion-capture system.

VI. Conclusions

In this work, we studied the effects that perception latency has on the maximum speed a robot can reach to safely navigate through an unknown environment. We provided a general analysis for a robot modeled as a linear second-order system with bounded inputs. We showed how the maximum latency the robot can tolerate to guarantee safety is related to the desired speed, the agility of the platform (e.g., the maximum acceleration it can produce), as well as other perception parameters (e.g., the sensing range). We compared frame-based cameras (monocular and stereo) against event cameras for quadrotor flight. Our analysis showed that the advantage of using an event camera is higher when the robot is particularly agile. We validated our study with experimental results on a quadrotor avoiding a ball thrown towards it a speeds up to $9\,\mathrm{m/s}$ using an event camera. Future work will investigate the use of event cameras for obstacle avoidance on a completely vision-based quadrotor platform, using on-board Visual-Inertial Odometry for state estimation.

Acknowledgments

We thank Henri Rebecq, Julien Kohler and Kevin Kleber for their help with the experiments.

References

[1] S. Karaman and E. Frazzoli, "High-speed flight in an ergodic forest," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2012.

[2] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014.

[3] C. Richter, W. Vega-Brown, and N. Roy, "Bayesian learning for safe high-speed navigation in unknown environments," in *International Symposium of Robotics Research, ISRR*, 2015, pp. 325–341.

[4] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, K. Karydis, N. Atanasov, G. Loianno, D. Scaramuzza, K. Daniilidis, C. J. Taylor, and V. Kumar, "Fast, autonomous flight in gpsdenied and cluttered environments," *J. Field Robot.*, vol. 35, no. 1, pp. 101–120, 4 2017.

[5] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems (RSS)*, July 2017.

[6] A. J. Barry, P. R. Florence, and R. Tedrake, "Highspeed autonomous obstacle avoidance with pushbroom stereo," *J. Field Robot.*, vol. 35, no. 1, pp. 52–68, 1 2018.

[7] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoingbased framework for the 2016 iros autonomous drone racing challenge," *J. Field Robot.*, vol. 35, no. 1, pp. 146–166, 5 2017.

[8] C. Greatwood, L. Bose, T. Richardson, W. Mayol, J. Chen, S. Carey, and P. Dudek, "Agile control of a uav by tracking with a parallel visual processor," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2017.

[9] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120dB 30mW asynchronous vision sensor that responds to relative intensity change," in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2006, pp. 2060–2069.

[10] M. Guo, J. Huang, and S. Chen, "Live demonstration: A 768x215; 640 pixels 200meps dynamic vision sensor," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017.

[11] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 15–22.

[12] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2007, pp. 3565–3572.

[13] T. Rosinol Vidal, H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. PP, no. 99, pp. 1–1, 2018.

[14] A. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[15] A. Handa, R. Newcombe, A. Angeli, and A. Davison, "Real-time camera tracking: When is high frame-rate best?" in *Eur. Conf. Comput. Vis. (ECCV)*, 2012.

[16] M. Vincze, "Dynamics and system performance of visual servoing," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1, 2000, pp. 644–649 vol.1.

[17] S. Behnke, A. Egorova, A. Gloye, R. Rojas, and M. Simon, "Predicting away robot control latency," in *RoboCup 2003: Robot Soccer World Cup VII*. Springer Berlin Heidelberg, 2004, pp. 712–719.

[18] P. Sermanet, R. Hadsell, J. Ben, A. Erkan, B. Flepp, U. Muller, and Y. LeCun, "Speed-range dilemmas for vision-based navigation in unstructured terrain," in *6th IFAC Symposium on Intelligent Autonomous Vehicles*, vol. 6, 2007, pp. 300–305.

[19] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 1, Sep 1994, pp. 314–321 vol.1.

[20] M. A. Henson and D. E. Seborg, Eds., *Nonlinear Process Control*. Prentice-Hall, Inc., 1997.

[21] D. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*, 2nd ed. Athena Scientific, 2005.

[22] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 2520–2525.

[23] R. Lozano, J. Guerrero, and N. Chopra, "Quadrotor flight formation control via positive realness multivehicle systems," 2012.

[24] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, 2015.

[25] E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing:learning agile flight in dynamic environments," *arXiv e-prints*, 2018. [Online]. Available: http://arxiv.org/abs/1806.08548

[26] T. Sayre-McCord, W. Guerra, A. Antonini, J. Arneberg, A. Brown, G. Cavalheiro, Y. Fang, A. Gorodetsky, D. McCoy, S. Quilter, F. Riether, E. Tal, Y. Terzioglu, L. Carlone, and S. Karaman, "Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[27] A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[28] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, 1996, pp. 226–231.

[29] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. Artificial Intell. (IJCAI)*, 1981, pp. 674–679.

Supplementary Material of:

How Fast is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid

Davide Falanga, Suseong Kim and Davide Scaramuzza

S1. Obstacle Avoidance: Brake or Avoid?

To avoid an obstacle, a robot can either stop before colliding or circumvent it by moving laterally. Fig. 1 shows a comparison between (i) the minimum time $t = \frac{\hat{\mathrm{v}}_1}{\bar{\mathrm{u}}_1}$ required for a robot to brake and stop before colliding, and (ii) the minimum time required to avoid the obstacle laterally without braking (see Sec. II-B2). We considered $\bar{\mathrm{u}}_1 = \bar{\mathrm{u}}_2 = 25\,\mathrm{m/s^2}$, and the horizontal axis reports the longitudinal speed towards the obstacle. The results show that the lateral avoidance maneuver requires less time at high speed, allowing faster navigation along the longitudinal axis. Additionally, a continuous motion along the desired direction is preferable over a *stop-avoid-go* behaviour, since would allow the robot to navigate faster and reach its goal earlier. Therefore, we consider only the case where the robot does not brake to prevent the collision, but rather executes a lateral avoidance maneuver.

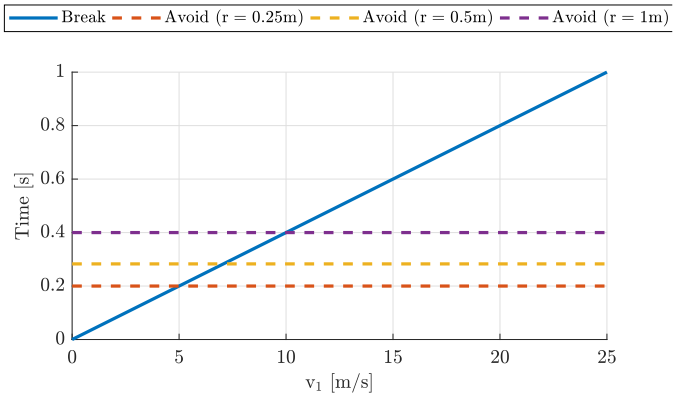

Fig. 1: Comparison between the minimum time required for a robot to completely stop before colliding (solid blue line) and the minimum time required to move laterally by an amount r (dashed lines), depending on the speed $\hat{\mathrm{v}}_1$ (horizontal axis).

S2. Sensitivity Analysis

Eq. (9) is particularly interesting for robot design to analyze what the best configuration in terms of perception and actuation systems is. As one can easily derive from (9), reducing the latency increases the maximum speed at which the robot can navigate the environment safely. However, it might not always be possible to reduce the sensing latency, or it might be better to change some other parameters of the system (e.g., the sensing range or the maximum acceleration), since this might produce better improvements at a lower cost. By performing a sensitivity analysis, we can study the impact of the sensing range, the latency, and the maximum input on the speed that the robot can reach.

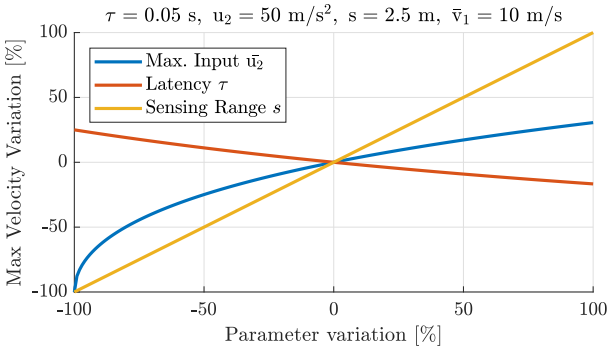

Fig. 2: Sensitivity of the maximum speed $\bar{\mathrm{v}}_1$ with respect to the perception (s, τ) and actuation ($\bar{\mathrm{u}}_2$) parameters of the system for the case $s = 2.5\,\mathrm{m}$, $\tau = 0.05\,\mathrm{s}$, $\bar{\mathrm{u}}_2 = 50\,\mathrm{m/s^2}$.

To do so, it is necessary to first define a set of parameters. For example, we consider the case $s = 2.5\,\mathrm{m}$, $\tau = 0.05\,\mathrm{s}$, $\bar{\mathrm{u}}_2 = 50\,\mathrm{m/s^2}$. This set of parameters, chosen as a representative case for the study in Sec. IV, according to (9) allow the robot to navigate at a maximum speed $\bar{\mathrm{v}}_1 = 10\,\mathrm{m/s}$. Based on these values, we vary each of the parameters while keeping the others constant to understand how the maximum speed the robot can achieve changes.

Fig. 2 shows the results of this numerical analysis for a variation of the parameters between -100% and 100% of the reference value (horizontal axis). On the vertical axis the percentage variation of $\bar{\mathrm{v}}_1$ is reported. As one can see, $\bar{\mathrm{v}}_1$ is very sensitive to the sensing range, whereas, except for extreme decreases of the maximum lateral acceleration (far left end of the blue line), the sensitivities with respect to $\bar{\mathrm{u}}_2$ and τ are comparable. However, it is not always possible to change the range of a sensing pipeline, whereas it could be possible to reduce its latency. This is the case, for example, of a DAVIS [1], a neuromorphic sensor comprising a frame and

an event camera sharing the same pixel array and optics. In such a case, it is possible to use frames or events depending on the need, but the sensing range, which depends on the sensor itself, cannot be modified for one modality without affecting the other. For this reason, in the remainder of this work will focus on the impact of the latency on the maximum speed a robot can navigate.

S3. Generalization to Multiple Obstacles

So far, we only considered the case where the robot faces a single obstacle and needs to avoid it. Although mathematically simple, our approach can generalize to multiple obstacles by iteratively running the same considerations previously described. Independently of the number of obstacles, we can always consider the closest obstacle to the robot along its direction of motion and perform the evaluation of Sec. II-B1 and II-B3. If the robot reaches a safe lateral position within the time to contact (3), we can consider the obstacle *avoided*, and the robot has to avoid the next obstacle along its path. The only difference with respect to the previously avoided obstacle is the distance between the obstacle and the robot along the longitudinal and lateral axes.

A conservative, yet effective analysis can be conducted for the case of navigation in environments with multiple obstacles by using our formulation under the following assumptions: (i) all the obstacles are considered to have the same size (i.e., the size of the largest obstacle); (ii) the distance between two consecutive obstacles along the longitudinal axis is sufficiently large to guarantee that the avoidance time in the case of no latency is lower than the time to contact.

S4. Monocular Frame-Based Camera

A. Sensing Range

For an obstacle to be detected with a frame-based camera, it has to occupy a sufficiently large number of pixels in the image. Let N be the number of pixels necessary to detect an obstacle. Furthermore, let α be the field of view of the sensor. Without loss of generality, we only consider the projection of an object along the horizontal axis of the camera, but similar results apply to the vertical axis.

Let q be the horizontal resolution of a camera. The angular resolution of the camera can be computed as $\theta = \frac{\alpha}{q}$. Let r_o be the size of an obstacle, d its distance to the camera, and assume it is placed such that the camera optical axis passes through its center (cf. Fig. 3). The obstacle spans an angle $\phi = 2\arctan\left(\frac{r_o}{2d}\right)$. For the obstacle to be visible in the image, it must be at a distance d such that $\phi = \theta$, which would result in a projection in the image of $1\,\text{pxl}$. However, $1\,\text{pxl}$ is typically not sufficient to detect an obstacle. Let N be the number of pixels one needs to detect an obstacle. For the obstacle to occupy at least N pixels in the image, we want that $\phi \geq N\theta$. We define the sensing range of a monocular camera s_M the maximum distance at which the obstacle is still detectable, namely the distance at which the previous condition is satisfied with the equality constraint:

$$s_\text{M} = \frac{r_o}{2\tan\left(\frac{N\theta}{2}\right)}. \tag{S.1}$$

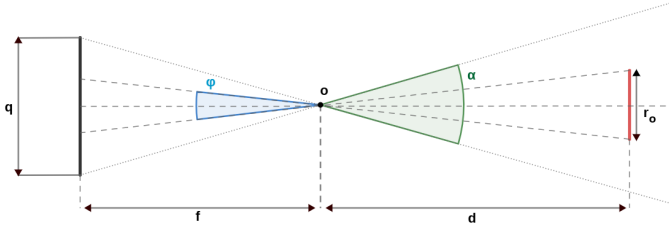

Fig. 3: A schematics representing the obstacle in front of the camera. The obstacle is represented in red, while the camera is in black on the left side of the image and has resolution q. The field of view α is highlighted in green, while the angle spanned by the obstacle in the image ϕ is highlighted in blue. d is the distance between the camera and the obstacle, while f is the focal length of the camera.

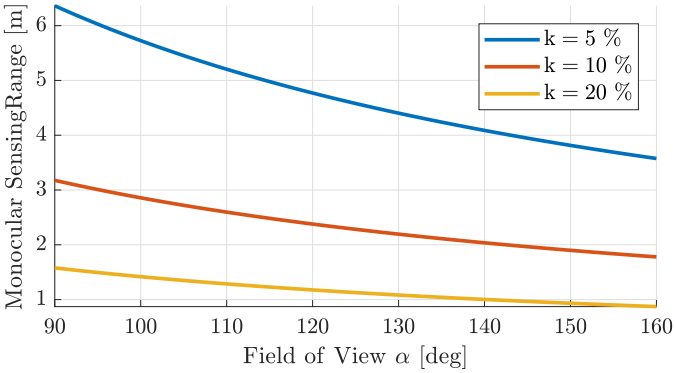

Fig. 4: The sensing range s_M for a monocular system depending on the field of view α. The the number of pixels N necessary to detect an obstacle of size $r_o = 0.5\,\text{m}$ are computed as a percentage k of the image resolution.

Eq. (S.1) shows that the sensing range of a monocular camera depends on its angular resolution θ. Fig. 4 shows the range at which a monocular system can detect an obstacle of size $r_o = 0.5\,\text{m}$ when this occupies a percentage $k = 5\%$, $k = 10\%$ and $k = 15\%$ of the image size q.

B. Latency

The latency of a camera-based perception system depends on (i) the time t_f between two consecutive images, (ii) the number of images necessary for detection, and (iii) the time to process each image. The first one only depends on the sensor itself, and includes, among the other things, the exposure time and the transfer time. The second and the third depend on the sensor, the computational power available and the algorithm used to detect the obstacle. It is therefore hard to provide an exact estimate of the actual latency of a perception system based on a monocular camera, since it depends on a large variety of factors. Thus, in this work we analyze its theoretical upper-bound and lower-bound to provide a *back-of-the-envelope* analysis of the possible performance achievable.

For a vision-based perception system to be effective, it has to produce its output in real-time. This means that, if n is the number of images necessary for the detection, the latter must happen before the frame $n + 1$ arrives. Therefore, the

frame-rate t_f of a camera provides an upper-bound for the latency of a monocular vision system. Assuming that 2 frames are sufficient to detect an obstacle along the robot's path, the latency for a monocular camera has an upper-bound given by $\tau_\mathrm{M} = 2t_\mathrm{f}$.

To have an estimate of the theoretical lower-bound of the latency of a frame-based camera, we neglect the processing time and only consider the delays caused by how such cameras work. More specifically, in the ideal case of negligible processing time, the lower-bound of the latency depends on (i) the time t_f between two consecutive triggers of the sensor, (ii) the exposure time t_E, and (iii) the time t_T necessary to transfer each frame. In the ideal case of no processing time, the latency of a frame-based camera has a lower-bound $\tau_\mathrm{M} = t_\mathrm{f} + t_\mathrm{T} + t_\mathrm{E}$. Typically, an image is transferred to the processing unit before the next one arrives, which means $0 < t_\mathrm{T} < t_\mathrm{f}$. The time t_T depends on the size of the image and the protocol used to communicate with the sensor. For example, a gray-scale VGA resolution image (i.e., $640 \times 480\,\mathrm{pxl}$) has a size of $2.1\,\mathrm{Mbit}$ and can be transferred in approximately $5\,\mathrm{ms}$ with a USB 2.0 connection ($480\,\mathrm{Mbit/s}$) and $0.4\,\mathrm{ms}$ with a USB 3.0 connection ($5\,\mathrm{Gbit/s}$). The exposure time depends on the amount of light available in the environment and cannot be larger than the time between two consecutive frames, i.e. $0 < t_\mathrm{E} < t_\mathrm{f}$.

S5. Stereo Frame-Based Camera

A. Sensing Range

Using stereo cameras, it is possible to triangulate points using only one measurement consisting of two frames grabbed at the same time. Let b be the baseline between the two cameras, f their focal length and l the disparity between the two images of a point of interest. The depth of such a point is given by $d = f\frac{b}{l}$. However, the uncertainty in the depth estimation ϵ_D grows proportional to the square of the distance between the camera and the scene [2], namely $\epsilon_\mathrm{D} = \frac{z^2}{bf}\epsilon_\mathrm{P}$, where ϵ_P is the uncertainty in the disparity matching. Therefore, we consider the sensing range for a stereo camera s_S as the maximum depth such that the uncertainty in the depth estimation is below a given percentage threshold k:

$$s_\mathrm{S} = \frac{kfb}{\epsilon_\mathrm{P}}. \tag{S.2}$$

Fig. 5 shows the sensing range of a stereo camera as a function of the baseline b such that the depth uncertainty ϵ_D is below 5% and 20% of the actual depth, for the cases of VGA ($640 \times 480\,\mathrm{pxl}$) and QVGA ($320 \times 240\,\mathrm{pxl}$) resolutions, assuming $\epsilon_\mathrm{P} = 1\,\mathrm{pxl}$.

B. Latency

Differently from monocular systems, stereo cameras capture simultaneously two frames using two cameras placed at a relative distance b. It is therefore possible to use a single measurement, i.e. two frames from two different cameras, to detect obstacles, for example computing the disparity between such frames, a depth map or an occupancy map. Depending on the technique used to detect obstacle using a stereo camera,

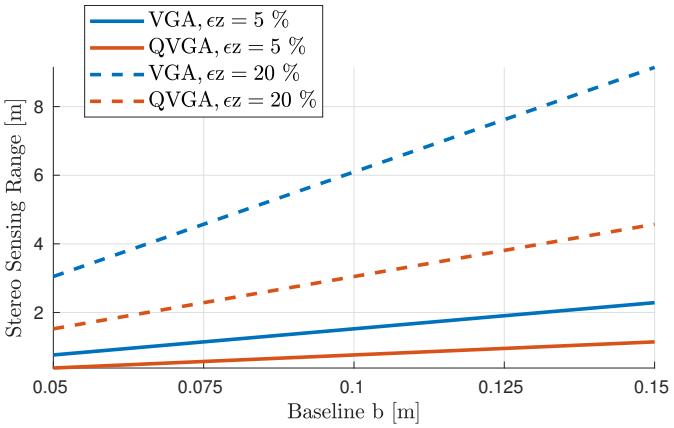

Fig. 5: The sensing range s_S for a stereo system depending on the baseline b for a focal length of $4\,\mathrm{mm}$. This sensing range guarantees that the uncertainty ϵ_D is below 5% and 20% of the actual depth.

the computational power available and the resolution of the output, the latency of a stereo system can vary significantly. For example, the Intel RealSense, provides a depth map at a frequency of $60\,\mathrm{Hz}$ (RealSense R200[1]), while the Bumblebee XB3[2] only provides its output at up to $16\,\mathrm{Hz}$. However, computing the latency of those measurements is not an easy task, since most of the commercially available sensors do not provide such information in their datasheets. An estimate of the latency of a wide variety of depth cameras is available thanks to the effort of the robotics community[3], according to which most of the stereo systems have a latency of one frame. Therefore, we consider as a lower-bound for stereo cameras the inverse of the frame-rate of the fastest sensor currently available on the market, namely the Intel RealSense, leading to a lower-bound $\tau_\mathrm{S} = 0.017\,\mathrm{s}$. For the upper-bound, instead, we can refer to the datasheet of the Stereolab ZED Mini[4], which has an estimated latency $\tau_\mathrm{S} = 0.07\,\mathrm{s}$.

S6. Monocular Event Camera

A. Sensing Range

Since monocular frame-based cameras and event camera often share the same sensor, we can use (S.1) to compute the sensing range of an event camera. However, the amount of pixels the obstacle must occupy in the image in order to be detected is significantly smaller. In principle, the obstacle would generate an event when each of its two edges occupy at least $1\,\mathrm{pxl}$ in the image. However, due to the noise of this sensor, the obstacle can be detected with an event camera when it occupies an amount of pixels in the image which is significantly larger than the amount of pixels it has to move to generate an event (see Sec. S6-B of this document). In this work, we assume that the obstacle size in the image must be at least one order of magnitude larger than the amount of pixels

[1] https://tinyurl.com/realsenser200
[2] https://tinyurl.com/bumblebeexb3
[3] https://rosindustrial.org/3d-camera-survey/
[4] https://www.stereolabs.com/zed-mini/

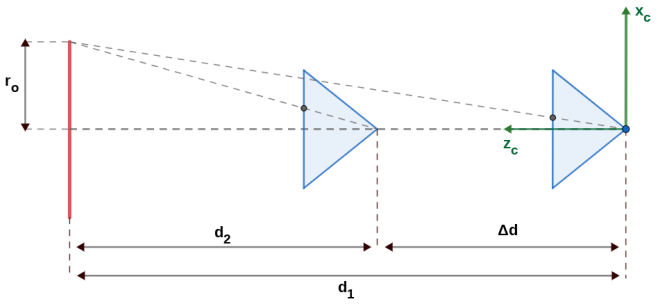

Fig. 6: A schematics representing the translation necessary for the obstacle to generate an event. The obstacle (in red on the left side of the picture) is projected on the image plane on a point which has horizontal component u depending on the distance d_i from the camera. The quantity Δd represents the distance the camera has to move such that the projection edge of the obstacle on the image plane moves by $1\,\mathrm{pxl}$.

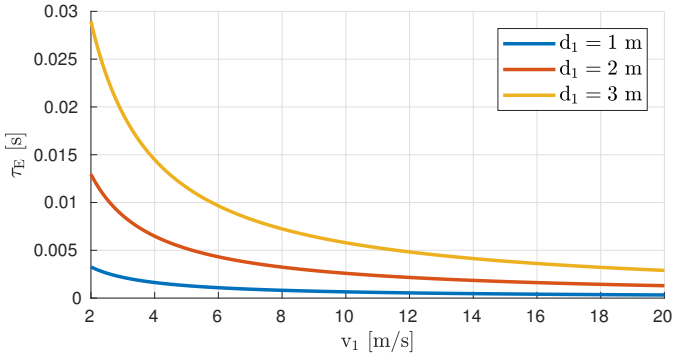

Fig. 7: The latency τ_{E} for an event camera depending on its distance from the obstacle d and the speed $\hat{\mathrm{v}}_1$. We considered the case of an event camera with a VGA resolution sensor and a focal length of $4\,\mathrm{mm}$.

it has to move to fire an event. Therefore, we compute the sensing range of an event camera using (S.1) with $N = 10$. This leads to a sensing range for an event camera which, depending on the field of view of the sensor, can span between $s_{\mathrm{E}} = 10\,\mathrm{m}$ and $s_{\mathrm{E}} = 20\,\mathrm{m}$ for an obstacle of size $r_o = 0.5\,\mathrm{m}$.

B. Latency

In this work, we assume that an obstacle can be detected using an event camera whenever its edges generate an event. For this to happen, there must be sufficient relative motion between the camera and the obstacle to cause a change of intensity sufficiently large to let an event fire. Typically, as shown in [3], the edge of an obstacle generates an event when its projection on the image plane moves by at least $1\,\mathrm{pxl}$. Without loss of generality, we analyze the horizontal motion of the obstacle in the image. Let d be the distance between the robot and the obstacle along the camera optical axis, and let r_o be the radius of the obstacle. Furthermore, assume the optical axis of the camera to pass through the geometric center of the obstacle, which we model here as a segment (cf. Fig. 6). The projection of a point p into the image plane has horizontal component u given by [4]:

$$u = f\frac{{}_cp_x}{{}_cp_z}, \tag{S.3}$$

where ${}_cp_x$ and ${}_cp_z$ are the components of p in the camera reference frame, and f is the camera focal length. In our case, ${}_cp_x = r_o$ and ${}_cp_z = d$. We can compute (S.3) for two values d_1 and $d_2 = d_1 - \Delta d$ of the distance along the optical axis, obtaining two different values u_1 and u_2, respectively. Equating $\Delta u = u_2 - u_1$ to the desired translation in the image plane necessary to generate an event (in our case, $\Delta u = 1\,\mathrm{pxl}$), we can compute the camera translation Δd as:

$$\Delta d = \frac{\Delta u d_1^2}{f r_o + \Delta u d_1}. \tag{S.4}$$

The time it takes the robot to cover such a distance Δd depends on its speed $\hat{\mathrm{v}}_1$:

$$\tau_{\mathrm{E}} = \frac{1}{\hat{\mathrm{v}}_1}\frac{\Delta u d_1^2}{f r_o + \Delta u d_1}. \tag{S.5}$$

Eq. (S.5) shows the time necessary to get an event from the edge of the aforementioned obstacle.

It is important to note that, since the transfer time for an event is in the order of a few microseconds [5], we consider it negligible. Similarly, we neglect the processing time for the case of event-based vision, since each pixel triggers asynchronously from the other and, therefore, the amount of data to be processed is significantly lower than the case where an entire frame has to be analyzed.

Fig. 7 shows the latency for an event camera (S.5) depending on its distance from the obstacle d and the speed $\hat{\mathrm{v}}_1$ in the case of VGA resolution and focal length of $4\,\mathrm{mm}$. It is clear that the theoretical latency of an event camera is not constant, but rather depends on the relative distance and the speed between the camera and the obstacle. Therefore, to compute the maximum latency that a robot can tolerate in order to safely navigate using an event camera, it is necessary to jointly consider the sensing range (Sec. III-B3), and Eq. (8) and (S.5). Intuitively speaking, this is due to the fact that event cameras are motion activated sensors. In order for the edges of an obstacle to generate an event, their projection in the image must move by at least $1\,\mathrm{pxl}$. For this to happen, the robot must move towards the obstacle by a quantity Δd which depends on its distance to the obstacle through (S.4). Therefore, the latency of an event camera, i.e. the time it takes the obstacle to generate an event, is given by the ratio between such a distance Δd and the robot's speed $\hat{\mathrm{v}}_1$, as shown by (S.5). However, the relative distance and speed between the robot and the obstacle also influence the time to contact (3), which must be larger than the avoidance time (4) for the robot the able to avoid the obstacle before colliding with it. Therefore, it is not possible to arbitrarily reduce the latency of an event camera for obstacle avoidance by increasing the robot's speed, since this might result in unfeasible avoidance maneuvers.

S7. Discussion

A. Stereo Frame or Monocular Event?

As shown in Tab. I, stereo cameras and event cameras provide results that, at least for currently available quadrotors, are comparable in terms of magnitude. Stereo cameras are currently still among the best options for autonomous quadrotor flight, since they provide a good compromise between latency and sensing range, without being very expensive. However, technological development in the event cameras might render them better solutions in the the future since (i) increasing the resolution would lead to higher angular resolution, which results in longer ranges, and (ii) they will be become cheaper as mass-production starts. Also, the sensing range of stereo cameras strongly depends on the baseline between the two cameras, which for small quadrotors are not always possible. Additionally, carrying one camera instead of two makes the platform lighter and, therefore, more agile [6]. Finally, event cameras have other advantages compared to frame-based cameras such as: (i) high dynamic range, which makes them more suited for navigation in adverse lighting conditions, where frame-based cameras might fail; (ii) their latency does not depend on the exposure time, which plays an important role in frame-based cameras and can significantly increase their latency; (iii) high temporal resolution, which reduces the motion blur and makes obstacle detection easier at high speed; (iv) low power consumption, which is desirable with small-scale robots [7].

B. Dynamic Obstacles

In this work, we only considered the case of navigation through static obstacles. Nevertheless, the mathematical framework provided in Sec. II can be used to consider the case of moving obstacles by taking into account that, in that case, the time to contact and the avoidance time depend on the relative distance and speed between the robot and the obstacle along the longitudinal and the lateral axes.

A fundamental assumption of our work is that the robot moves along a direction which makes the obstacle detectable and eventually leads to a collision. In the case of moving obstacles, this might not always be the case. Indeed, depending on the relative distance and speed between the robot and the obstacle, a number of cases can occur: (i) the robot detects the obstacle, but their relative motion does not lead to a collision; (ii) the robot detects the obstacle, and their relative motion leads to a collision; (iii) the robot cannot detect the obstacle, and their relative motion does not lead to a collision; (iv) the robot cannot detect the obstacle, but their relative motion leads to a collision. It is clear that, in the case of moving obstacle, the amount of cases to be taken into account and the parameters to be considered increases significantly. For example, the field of view of the robot also plays a crucial role in the case of moving obstacles. Indeed, for a given relative speed, depending on the field of view of the sensing pipeline it is equipped with, the robot might or might not be able to detect the obstacle. In the case it is able to detect the obstacle, the relative distance at the moment the latter enters the sensing range depends on how large the field of view is, which then determines the time to contact. Therefore, in the case of a robot navigating through moving obstacles, a broader and more detailed analysis of the dependence of the maximum achievable speed on each parameter is necessary.

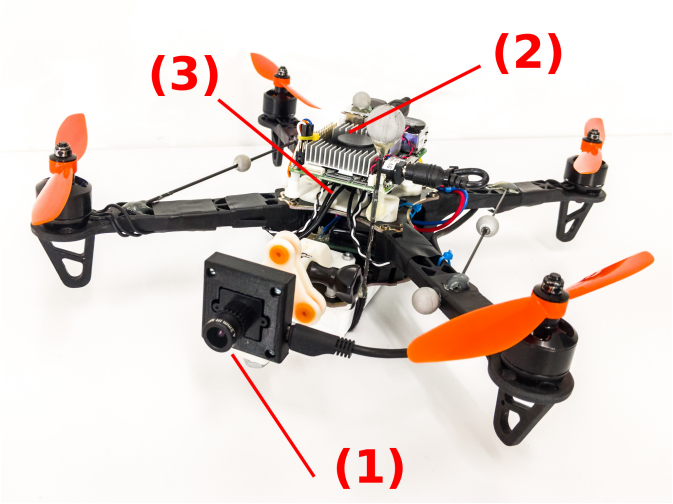

Fig. 8: The quadrotor used for the experiments. (1) The Insigthness SEEM1 sensor. (2) The Intel Upboard computer, running the detection algorithm. (3) The Lumenier F4 AIO flight controller, receiving commands from the ground station.

Intuitively, moving obstacles would highlight the benefits of event cameras against other sensors. To compute the latency of an event camera, we considered the case of a robot moving towards a static obstacle, placed in the center of the image, along a direction parallel to the camera's optical axis. This represents a sort of *worst case* for event cameras, since the apparent motion between the sensor and the obstacle is small. Conversely, an obstacle moving along the lateral axis would increase the apparent motion in the image and, therefore, generate an event earlier than in the case of static obstacles. Additionally, when obstacles enter the sensing area at a short distance, the importance of latency increases as the time to contact decreases. For this reason, we expect that event cameras would allow faster flight in the case of moving obstacles, especially for short sensing ranges (or, equivalently, for obstacles entering the sensing area at short distances). We are currently working on analyzing the impact of the sensing pipeline's parameters (latency, sensing range and field of view) for the case of moving obstacles from a mathematical point of view.

S8. Experiments

A. Experimental Platform

We used a custom-made quadrotor platform to perform the experiments. The vehicle was built using the DJI F330 frame, and was equipped with Cobra CM2208 motors and Dalprop 6045 propellers. The tip-to-tip diagonal of the quadrotor was $50\,\mathrm{cm}$, with an overall take-off weight of approximately $860\,\mathrm{g}$ and a thrust-to-weight ratio of roughly 3.5. We used an Optitrack motion-capture system to measure the state of the quadrotor, as well as the position and velocity of the ball.

s [m]	μ [s]	σ [s]
1	0.0037	0.0030
2	0.0688	0.0474
3	0.1832	0.0766

TABLE I: The mean μ and standard deviation σ of the latency for the obstacle detection algorithm proposed in this work based on the Insightness SEEM1 sensor.

The ball measurements were not used by the vehicle, which only relied on the information coming from the onboard obstacle detection algorithm, and were used as ground truth to benchmark the sensing pipeline. To detect the obstacle, we mounted an Insightness SEEM1 [5] neuromorphic sensor looking forward, and an Intel UpBoard computer running the obstacle detection algorithm described in the previous section. The horizontal field of view of the sensor was approximately $90°$. Whenever the obstacle was detected, a trigger signal was sent to a ground-station computer connected to the motion-capture system and running the control stack described in [8], which then initiated the avoidance maneuver. The control commands (i.e., collective thrust and body rates) were sent to a Lumenier F4 AIO flight controller by the ground-station through a Laird RM024 radio module.

B. Obstacle Detection with an Event Camera: Theoretical and Practical Latency

As described in Sec. V of the main manuscript, we performed actual experiment on a quadrotor equipped with an Insightness SEEM1 sensor having QVGA resolution (i.e., $320 \times 240\,\text{pxl}$). We estimated that, in order to obtain reliable measurements of the obstacle, a displacement Δu of $5\,\text{pxl}$ was typically necessary. Fig. 9 shows the theoretical latency of such a sensor for obstacle detection, according to the model proposed in Sec. S6-B, for a sensing range of $1\,\text{m}$, $2\,\text{m}$ and $3\,\text{m}$, with $\Delta u = 5\,\text{pxl}$.

To validate these results, we performed a quantitative analysis using ground truth data provided by an Optitrack motion-capture system. More specifically, we performed 100 experiments throwing the ball, anchored to a table through a leash to prevent collisions, towards the quadrotor, and used data from the motion-capture system to measure the moment when the ball entered the sensing range s of the camera. This was manually set to three different values, i.e. $s = 1\,\text{m}$, $s = 2\,\text{m}$ and $s = 3\,\text{m}$. For each of these values, we computed the time when the sensing pipeline detected the ball for the first time, and compared it to the time when the obstacle actually entered the sensing range using data from the motion-capture system. This comparison allowed us to estimate the latency of our event-based obstacle detection algorithm, and the results are shown in Fig. 10 for a range of obstacle speeds between $5\,\text{m/s}$ and $9\,\text{m/s}$.

[5]http://www.insightness.com/technology

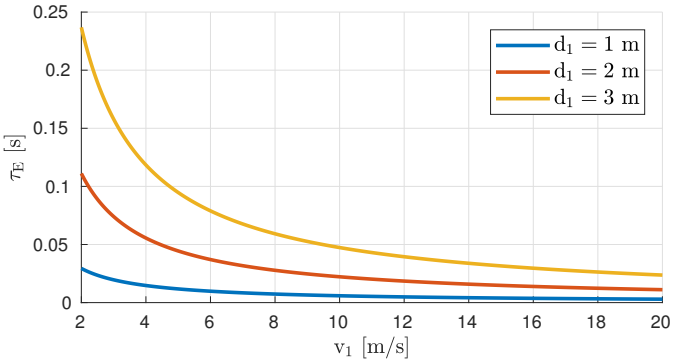

Fig. 9: The theoretical latency τ_E for the Insigthness SEEM1 used in our experiments, depending on its distance from the obstacle d and the speed $\hat{v}_1$. We considered the case of an event camera with a QVGA resolution sensor and a focal length of $4\,\text{mm}$.

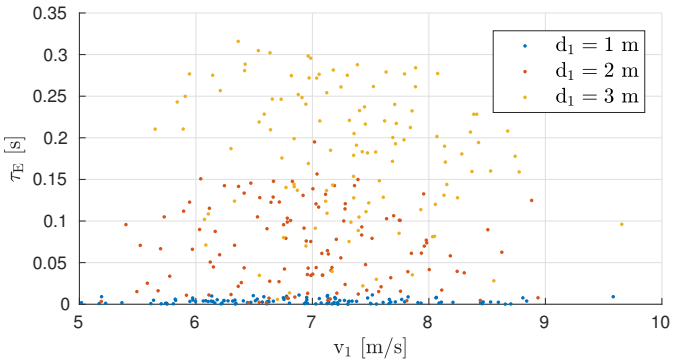

Fig. 10: The measured latency τ_E of our event-based obstacle detection algorithm using an Insightness SEEM1, depending on its distance from the obstacle d and the speed $\hat{v}_1$.

C. Obstacle Detection with an Event Camera: Discrepancy Between Theory and Practice

As one can notice, the results in Fig. 10 agree with the theoretical lower-bound of the latency expected for the sensor used in our experiments, shown in Fig. 9. Tab. I reports the mean μ and standard deviation σ of the latency of our event-based obstacle detection algorithm, depending on the desired sensing range. As the sensing range increases, also the error between the mean value and the expected theoretical latency increases. Similarly, the standard deviation becomes larger. We believe this effects to be mainly due to two factors.

First, the output of an event camera is particularly noisy. The higher the noise level, the larger the amount of events that need to be processed and, therefore, the higher the computational cost of our algorithm. In our case, the noise comes from both actual sensor noise and events generated by the static part of the scene which are not perfectly compensated by our algorithm.

Second, the resolution of our sensor is particularly low. This has a twofold consequence. The first is that the size of the obstacle in the image is not very large when it is far away from the camera. The second is that, when the obstacle is far from the camera, it needs to move by a significant amount in

order for its projection in the image to move by an amount $\Delta u = 5\,\text{pxl}$. The closer it gets to the camera, the smaller the distance it has to travel to produce such displacement Δu, which, for a constant velocity of the obstacle, translates into a lower detection latency. Additionally, when the obstacle is close to the camera, it occupies a significant portion of the image, making its detection easier.

Therefore, as the sensing range increases, the difference between the theoretical model (Sec. S6-B) and the actual sensing pipeline becomes more and more important. However, near-future improved versions of event-based sensors can bridge this gap and render event-based obstacle detection pipelines closer to the theoretical model we propose in this work. More specifically, we believe that event cameras with higher resolution could lead to better and faster obstacle detection pipelines. An additional benefit of large resolutions is the possibility of mounting lenses providing larger field of views, which are desirable to sense obstacles, without sacrificing the angular resolution of the sensor.

REFERENCES

[1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10):2333–2341, 2014.

[2] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Variable baseline/resolution stereo. In *IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2008.

[3] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4874–4881, 2015.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[5] Elias Mueggler, Nathan Baumli, Flavio Fontana, and Davide Scaramuzza. Towards evasive maneuvers with quadrotors using dynamic vision sensors. In *Eur. Conf. Mobile Robots (ECMR)*, pages 1–8, 2015.

[6] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *J. Field Robot.*, 31(11), September 2012.

[7] Daniele Palossi, Antonio Loquercio, Francesco Conti, Eric Flamand, Davide Scaramuzza, and Luca Benini. Ultra low power deep-learning-powered autonomous nano drones. *arXiv e-prints*, 2018.

[8] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robot. Autom. Lett.*, 3(2):620–626, April 2018.