Direct Time-to-Contact Estimation for Unmanned Aerial Vehicle Control

Zhang-Wei Hong
zwhong@mit.edu

Tsun-Hsuan Wang
tsunw@mit.edu

Abstract

Time-to-contact (TTC) is a useful measurement for drone control since TTC gives the proximity estimation to obstacles or ground. With TTC, the feedback controller can then adjust the motion before the collision. However, most of the TTC-based control methods cannot be deployed to drones on a scale since either expensive sensors or costly computations are required. Toward deploying drones on a scale, a low-cost and efficient TTC-based control is proposed. This method directly estimates TTC from two consecutive monocular camera images without expensive sensors and complicated computations. To validate the effectiveness of the proposed method, experiments in several different imaging conditions are conducted. The experimental results show that the proposed method can perform smooth landing, as does the TTC-based controller with ground truth TTC. The code is available at `https://github.com/zswang666/mv_final`.

1. Introduction

Time-to-contact (TTC) [6] plays an important role in autonomous vehicle control such as self-driving cars [15] or drones [14, 18, 11]. The proximity to objects given by TTC serves as informative feedback for the control system to control the vehicle. By tweaking the velocity in proportion to the TTC estimation, vehicles can decelerate smoothly to the rest. However, typical TTC-based methods are not suitable for unmanned aerial vehicles (UAVs) since UAVs are expected to be deployed on low-power computing devices in scale.

Typical TTC estimation methods come at a high cost and are unscalable since expensive sensors and intensive computations are required. Common methods measure TTC via taking time derivative over the spatial information from GPS [11, 5] or optical flow [3, 9, 10]. The high cost hinders GPS-based solutions from large scale deployment; costly computations [1] in deriving TTC from optical flows inhibits the optical flow based methods from being applied in low-power computing devices. A cheaper and efficient TTC estimation method that can execute on low-power computing devices is necessary toward large scale deployment of TTC-based control systems on UAVs.

This paper proposes a low-cost yet efficient feedback control method based on direct TTC estimation [8] from a monocular camera. The direct TTC estimation relaxes the need for expensive sensors and costly computations. Rather than taking time derivative on the estimated spatial information [11] or optical flow [10], TTC is derived from the brightness gradients between two consecutive images. With this direct method, TTC-based control can be performed solely by a cheap monocular camera. Furthermore, calibration is unnecessary for the proposed method since extrinsic and intrinsic camera parameters are not needed for the direct TTC estimation method.

To validate the proposed method's effectiveness, the simulated experiments are conducted in a range of different scenarios. The scenarios are designed to test UAV control's robustness in several challenging conditions, including shadowy weather, dusty environment, dynamic lighting, and stormy weather. The experiments are focused on testing the landing control performance since landing is one of the prominent applications of TTC-based control [5]. The experimental results show that the proposed method can exhibit smooth landing, as does the TTC-based control, which accesses the ground truth TTC, suggesting that it can be a scalable alternative for UAV control.

The rest of this paper is structured as the follows. Section 2 lists and reviews the relevant literature. Section 3 presents the essential preliminaries. Section 4 explains the implementation of the proposed method. Section 5 discusses the experimental results. Section 6 concludes this paper.

2. Related works

TTC has been widely used in robotics [16] and vehicle control for reactive motion control (e.g., landing [18, 11, 14], docking [17], and driving [15, 12, 2]). The closest work [15] uses the TTC directly estimated from the image brightness to control cars' for avoiding collision to front cars. The works [2, 12] on robotics perform obstacle avoidance based on the TTC to obstacles. In addition to obstacle avoidance, the work [17] exploits the TTC to dock the

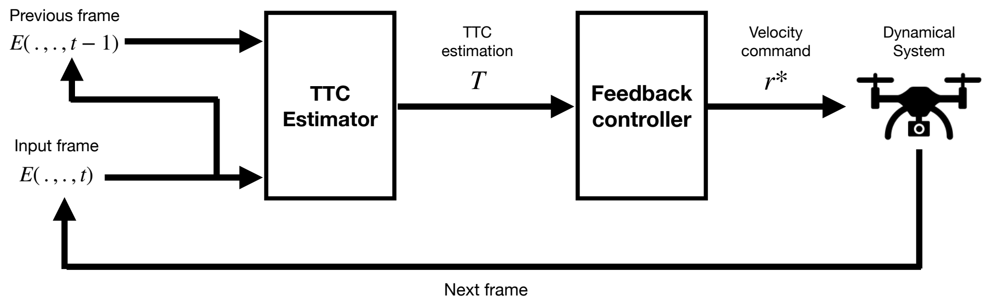


Figure 1: The overview of the proposed method. The TTC estimator predicts the TTC by the two consecutive images. The feedback controller gives the velocity command based on the TTC estimation.

wheeled robot smoothly. UAV landing, which is the focus of this paper, is also one of TTC's prominent applications. Several works [18, 11, 14, 5] use TTC to derive control laws for landing on the ground. However, most of the previous works estimate TTC by unscalable and computationally intensive solutions.

The TTC estimation methods used in the above works are mostly based on GPS and costly computations. GPS-based methods [11] are expensive and unscalable and noisy since GPS is inapplicable in indoor cases, and the measurement is easily perturbed by the environment. Optical flow based methods [3] require visual feature extraction [10, 4, 13, 9], object segmentation [10], and iterative computations [1] for optical flows. These computations in optical flow based methods are too expensive to deploy on most of the low-power devices on drones.

3. Background

This section introduces direct TTC estimation method [8]. Since this paper focuses on UAV landing problem, translational motion relative to the ground which is perpendicular to camera's optical axis is considered.

The motion involved in landing can be described in two coordinate system: camera-centric world coordinate system (X, Y, Z) and image plane coordinate system (x, y). For the world one, the origin locates at the center of projection (COP) of the camera and the axes follow the right-hand rule where z-axis points toward the scene. The origin of the image plane one is at the center of the image. TTC is defined as the ratio of the distance to the instantaneous velocity. The expression of TTC associated to the coordinate system can be written as

$$T = -\frac{Z(t)}{\frac{dZ(t)}{dt}} = -\frac{Z(t)}{W(t)},$$

where $Z(t)$ is the distance from the COP to the plane (e.g., obstacle or ground) and $W(t)$ (be negative when the object is approaching the camera) is the instantaneous velocity at time t. Hereinafter, to express the coordinate at time t, the expression of coordinates are augmented as $(x(t), y(t))$ and $(X(t), Y(t), Z(t))$.

To recover T from image brightness, the constant brightness constraint [7]

$$\frac{dE(x(t), y(t), t)}{dt} = uE_x + vE_y + E_t = 0 \qquad (1)$$

is employed, where $E(x(t), y(t), t)$ is the image brightness value at (x, y) and time t, $(u, v) = (\frac{dx(t)}{dt}, \frac{dy(t)}{dt})$. E_x, E_y, and E_t are the partial derivatives of brightness with respect to x, y at (x, y) and t. The constant brightness constraint can be related to TTC by differentiating the perspective projection equation

$$\frac{x(t)}{f} = \frac{X(t)}{Z(t)} \text{ and } \frac{y(t)}{f} = \frac{Y(t)}{Z(t)}$$

with respect to t at both sides

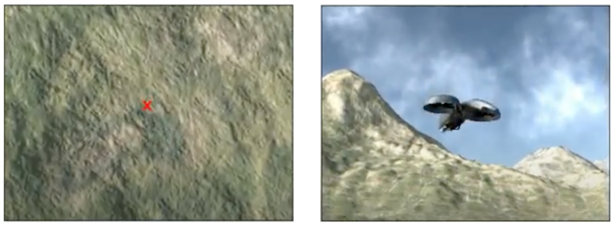
$$\begin{aligned} \frac{1}{f}\frac{dx(t)}{dt} &= \frac{u(t)}{f} = \frac{U(t)}{Z(t)} - \frac{X(t)}{Z(t)}\frac{W(t)}{Z(t)} = \frac{U(t)}{Z(t)} - \frac{x(t)}{f}\frac{W(t)}{Z(t)} \\ \frac{1}{f}\frac{dy(t)}{dt} &= \frac{v(t)}{f} = \frac{V(t)}{Z(t)} - \frac{Y(t)}{Z(t)}\frac{W(t)}{Z(t)} = \frac{V(t)}{Z(t)} - \frac{y(t)}{f}\frac{W(t)}{Z(t)}, \end{aligned} \qquad (2)$$

where $(U, V, W) = (\frac{dX(t)}{dt}, \frac{dY(t)}{dt}, \frac{dZ(t)}{dt})$ correspond to instantaneous velocities at time t of a point in the world coordinate system and f is the focal length of the camera. Plugging the expressions of u and v (Equation (2)) to Equation (1) and regrouping terms lead to

$$AE_x + BE_y + CG + E_t = 0, \qquad (3)$$

where $A = f\frac{U(t)}{Z(t)}$, $B = f\frac{V(t)}{Z(t)}$, $C = \frac{-W(t)}{Z(t)}$, and $G = (x(t)E_x + y(t)E_y)$. TTC can then be obtained by solving the following least-square problem associated with A, B, and C

$$\min_{A,B,C} \sum_{x,y} (AE_x + BE_y + CG + E_t)^2. \qquad (4)$$



(a) UAV's View (b) Third-Person View

Figure 2: Illustration of UAV landing simulation.

Given the closed-form solutions [8] to A, B, and C, TTC can then be derived as the follows

$$T = 1/C.$$

4. Method

4.1. Overview

The overall architecture of the proposed method is illustrated in Fig. 1. The input frame $E(.,.,t)$ and the previous frame $E(.,.,t-1)$ are forwarded into the TTC estimator, where t corresponds to a discrete timestep[1]. The TTC estimator uses the direct method introduced in Section 3 to estimate TTC T. The feedback controller takes T to compute the velocity command r^* for the drone at the timestep t. After the drone executes the velocity command, the next frame is feed into the TTC estimator. The loop of control terminates once the landing criteria (later specified in the Section 5) is met.

4.2. Direct TTC estimation based Feedback Control for Landing

The goal of landing is to descent the drone along a desired trajectory which can be expressed as

$$d(t) = \frac{1}{2}at^2 \ \forall t < 0 \tag{5}$$

where $d(t)$ is the distance from the drone to the ground at time t and a is a given constant acceleration downward. For a desired trajectory, $d(0) = 0$ and $d(t) > 0, t < 0$. To produce the control command that produces this trajectory, Equation 5 is related to TTC as follows

$$T = -\frac{d(t)}{r(t)} = -\frac{\frac{1}{2}at^2}{at} = -\frac{1}{2}t, \tag{6}$$

where $r(t)$ is the instantaneous velocity corresponding to the desired trajectory at time t. As Equation (6) is invertible, $r(t)$ can then be connected to TTC as the follows

$$r(t) = at = -2aT, \tag{7}$$

[1]All timesteps have the fixed delta real time to other timesteps.

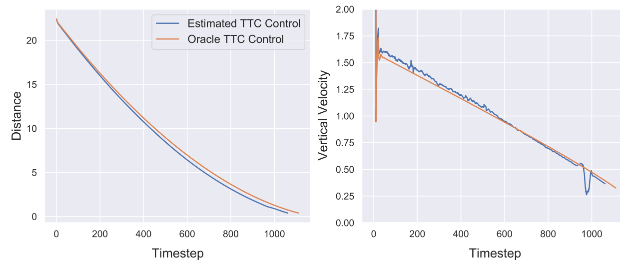


Figure 3: Control with estimated and ground-truth TTC.

where $t = -2T$. Combining the T estimated by the TTC estimator in Fig. 1, the desired velocity command r^* issued to the drone's dynamical system can then be solved by Equation (7). Moreover, this control law can perform safe landing which has small impact force to the ground. As $T \to 0$ when approaching the ground, this control law (Equation (7)) decreases the velocity to zero smoothly in descending, leading to smooth and safe landing.

Note that in the drone's dynamical system, the low-level controller can tweak the drone's velocity to the desired velocity and compensate the external force (e.g., gravity) in a negligible delta time.

4.3. Implementation

Each frame is converted to grayscale to get the brightness values $E(.,.,t)$. As noise in large images is inevitable, each frame is downsampled by a 5×5 block average filter followed by a low pass filter. The resultant frame is smaller and less noisy. Next, the brightness gradients at each image position (x, y) are approximated as the follows

$$\begin{aligned} E_x(x,y,t) &= E(x+1,y,t) - E(x,y,t) \\ E_y(x,y,t) &= E(x,y+1,t) - E(x,y,t) \\ E_t(x,y,t) &= E(x,y,t) - E(x,y,t-1), \end{aligned}$$

where x and y correspond to the pixel position on the image.

5. Experiments

5.1. Testing Environment

To examine the performance of our TTC-based control, we create a virtual environment for UAV landing in Unity[2]. Working in simulation allows access to ground truth state of the vehicle/environment and the ability to simulate various weather conditions that might affect the behavior of the controller. There are a RGB camera and a 1-beam range sensor equipped at the bottom of the UAV, facing toward ground. The UAV takes colored images as input to perform TTC-based control and once descend to a sufficiently low height (within the operational domain of the range sensor, $0.4m$ in our experiment), TTC-based control terminates and the

[2]unity.com/

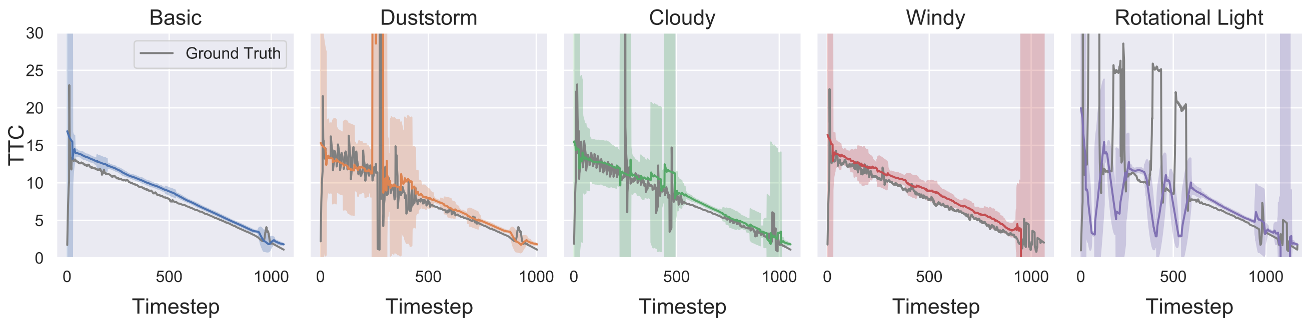


Figure 4: TTC estimation at different weather condition. (The shaded area indicates running standard deviation)

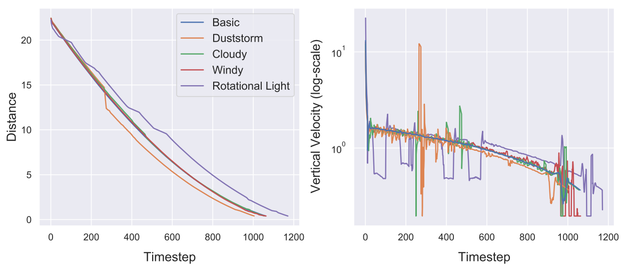


Figure 5: Distance and vertical velocity through time.

landing task is considered completed. The vehicle dynamics follows simple rigid body dynamics with 1-dimensional velocity control along yaw axis (vertical velocity). Fig. 2 is an illustration of the environment.

Apart from no external interference on the UAV (*basic*), we implement four different weather conditions, including *dust storm*, *cloudy*, *windy*, *rotational light*. *Dust storm* and *cloudy* effectively introduce dynamic objects to the image. *Windy* introduces additional random force other than that along the optical axis. Finally, *rotational light* resembles time-varying lighting condition.

5.2. Experimental Results

In Fig. 3, we verify the behavior of controller by comparing between using estimated and oracle TTC, where the latter one can be computed by dividing distance to the ground (unknown to the UAV) by the vertical velocity. Control with estimated TTC yields more jittering results and the impulse at the end results from shadow of UAV itself when approaching the ground. In addition, we compare terminal speed between landing with and without TTC ($0.36m/s$ vs. $20.80m/s$). We further test our controller on various weather conditions. We show the performance of TTC estimation in Fig. 4. We compute running mean and standard deviation with window size equal to 50 steps. First we point out every weather conditions suffers more or less from unstable estimation at the end since UAV creates shadow of itself when extremely close to the ground, giving inconsistent E_t. In *dust storm*, most jittering occurs at the first half throughout the landing because the moving dust that disturbs E_t estimates gets thinner (less effective) while the UAV approaching the ground. In *cloudy*, the unsteadiness depends on cloud shadow movement and may spreads randomly throughout the landing. In *windy*, we can spot similar but slightly more unstable behavior in comparison to *basic*, implying TTC estimation capable of handling arbitrary translation, except for the end where incorrect E_t drastically magnifies the error while jointly taking non-vertical translation into consideration.

In *rotational light*, the estimation not only is extremely unstable but has large errors (e.g., at ~ 500 steps). This is due to time-varying lighting condition affects a huge or even all regions in the image as opposed to effects on local region in *cloudy* case. Besides, the lighting change is much more structured than *dust storm*, which more resembles adding noise to the image. Accordingly, direct method will be not only subject to inconsistency from (effectively) dynamic objects in the scene but misled to an erroneous solution. Finally, in Fig. 5, we compare distance to the ground and vertical velocity throughout the landing in various weather condition. Despite large variance in TTC estimate, all variants have similar slope in distance since the jitters in velocity will be smoothed out after integration for distance. For velocity, we can observe similar trend to TTC estimate, e.g., in *rotational light* (purple). This is as expected since velocity control is linearly related to TTC, shown in Equation (7). This result emphasizes the importance of stable TTC estimate for achieving well-behaved controller.

6. Conclusion

We build a cheap and efficient UAV landing controller based on direct TTC estimation that can handle arbitrary translation of the vehicle and ground plane perpendicular to the optical axis. We show that estimated TTC can be used as a feedback signal for velocity control to achieve safe landing. To demonstrate the effectiveness of the controller, we build a simulation environment for UAV landing. We further test the landing control under various weather conditions and investigate failure modes of the method. Overall, the experimental results suggest a promising direction of direct TTC-based control for UAV landing.

Workload Distribution

- Zhang-Wei Hong (50%):

  TTC + Controller + Experiments

- Tsun-Hsuan Wang (50%):

  Simulation + Controller + Experiments

References

[1] Guillem Alenyà, Amaury Nègre, and James L Crowley. A comparison of three methods for measure of time to contact. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4565–4570. IEEE, 2009.

[2] Guillmem Alenya, Amaury Nègre, and James L Crowley. Time to contact for obstacle avoidance. 2009.

[3] TA Camus. Calculating time-to-contact using real-time quantized optical flow. 1995.

[4] GCHE De Croon, D Izzo, and G Schiavone. Time-to-contact estimation in landing scenarios using feature scales. *Acta Futura*, 5:73–82, 2012.

[5] Alvika Gautam, PB Sujit, and Srikanth Saripalli. A survey of autonomous landing techniques for uavs. In *2014 international conference on unmanned aircraft systems (ICUAS)*, pages 1210–1218. IEEE, 2014.

[6] Heiko Hecht and Geert Savelsbergh. *Time-to-contact*, volume 135. Elsevier, 2004.

[7] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[8] B. K. P. Horn, Y. Fang, and I. Masaki. Time to contact relative to a planar surface. In *2007 IEEE Intelligent Vehicles Symposium*, pages 68–74, 2007.

[9] Dario Izzo and Guido De Croon. Landing with time-to-contact and ventral optic flow estimates. *Journal of Guidance, Control, and Dynamics*, 35(4):1362–1367, 2012.

[10] Yukimasa Kaneta, Yuki Hagisaka, and Kazuyuki Ito. Determination of time to contact and application to timing control of mobile robot. In *2010 IEEE international conference on robotics and biomimetics*, pages 161–166. IEEE, 2010.

[11] Farid Kendoul. Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems. *The International Journal of Robotics Research*, 33(2):237–267, 2014.

[12] Olivier Koch, Matthew R Walter, Albert S Huang, and Seth Teller. Ground robot navigation using uncalibrated cameras. In *2010 IEEE International Conference on Robotics and Automation*, pages 2423–2430. IEEE, 2010.

[13] Angel J Sanchez-Garcia, Homero V Rios-Figueroa, Hugues Garnier, Gustavo Quintana-Carapia, Ericka Janet Rechy-Ramirez, and Antonio Marin-Hernandez. Predicting collisions: time-to-contact forecasting based on probabilistic segmentation and system identification. *Advanced Robotics*, 32(8):426–442, 2018.

[14] Benjamin Thomsen, Mingfeng Zhang, and Inna Sharf. Bio-inspired time-to-contact control for autonomous quadrotor vehicles. In *AIAA Guidance, Navigation, and Control Conference*, page 0384, 2016.

[15] Liang Wang and Berthold KP Horn. Time-to-contact control: improving safety and reliability of autonomous vehicles. *International Journal of Bio-Inspired Computation*, 16(2):68–78, 2020.

[16] Haijie Zhang, Bo Cheng, and Jianguo Zhao. Extended tau theory for robot motion control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5321–5326. IEEE, 2017.

[17] Haijie Zhang and Jianguo Zhao. Biologically inspired vision based control using featureless time-to-contact estimations. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1133–1138. IEEE, 2016.

[18] Zhen Zhang, Pu Xie, and Ou Ma. Bio-inspired trajectory generation for uav perching movement based on tau theory. *International Journal of Advanced Robotic Systems*, 11(9):141, 2014.