

Shot Type Constraints in UAV Cinematography For Autonomous Target Tracking

Iason Karakostas*, Ioannis Mademlis*, Nikos Nikolaidis and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract

During the past years, camera-equipped Unmanned Aerial Vehicles (UAVs) have revolutionized aerial cinematography, allowing easy acquisition of impressive footage. In this context, autonomous functionalities based on machine learning and computer vision modules are gaining ground. During live coverage of outdoor events, an autonomous UAV may visually track and follow a specific target of interest, under a specific desired shot type, mainly adjusted by choosing appropriate focal length and UAV/camera trajectory relative to the target. However, the selected UAV/camera trajectory and the object tracker requirements (which impose limits on the maximum allowable focal length) affect the range of feasible shot types, thus constraining cinematography planning. Therefore, this paper explores the interplay between cinematography and computer vision in the area of autonomous UAV filming. UAV target-tracking trajectories are formalized and geometrically modeled, so as to analytically compute maximum allowable focal length per scenario, to avoid 2D visual tracker failure. Based on this constraint, formulas for estimating the appropriate focal length to achieve the desired shot type in each situation are extracted, so as to determine shot feasibility. Such rules can be embedded into practical UAV intelligent shooting systems, in order to enhance their robustness by facilitating on-the-fly adjustment of the cinematography plan.

Keywords: UAV cinematography, shot type, target tracking, autonomous drones

1. Introduction

Automation in applications involving cinematic video footage (e.g., TV/movie production, outdoor event coverage, advertising, etc.) is constantly improving, both in the post-production stage (e.g., shot cut/scene change detection [26], automated editing [3] or framing [1], etc.) and during production (e.g., [6]). Relevant algorithms typically

^{1*}The first two authors contributed equally and are joint first authors.

²2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

utilize expert knowledge about the film creative process and the cinematic grammar, in order to assist in footage shooting, indexing, annotation, and/or post-processing.

While filming, the most important creative decisions made by the director pertain to the shot type and the camera motion type. The shot type is defined mainly by the percentage of the video frame area covered by the target being filmed. In traditional film grammar the target is assumed to be a human subject, but this is not strictly necessary (for instance, it can be a static or moving vehicle). If the distance between the target and the camera remains constant, the shot type is controlled primarily by changing the camera focal length f , hence adjusting the zoom level. The camera motion type refers to the camera motion trajectory relative to the target for the duration of a shot.

Despite the presence of a large body of research dedicated to automated shot type and camera motion type recognition in existing footage during post-production (e.g., [37] [4] [11] [8]), little work has been performed on autonomously capturing new videos with desired shot type/camera motion type combinations. Such methods are typically given the label of intelligent shooting. In dynamic environments, relevant approaches require robotic cameras that partially rely on real-time machine learning and computer vision algorithms, for visually detecting/tracking [25] [38] [19] [27] [31] [32] and physically following a specific desired target (e.g., the lead athlete in a race). However, to the best of our knowledge, the interplay between 2D visual tracker operation and cinematographic properties, i.e., shot type and camera motion type, has not been thoroughly investigated.

An important issue from this respect is determining the range of feasible shot types at each time point, so that visual tracking algorithms do not fail. The selected shot type severely affects the perceived 2D displacement of a moving target image between consecutive video frames, due to the effects of zooming. Thus, real-time visual object tracking [18] is heavily influenced by cinematography decisions, given that virtually all trackers search a restricted video frame region for the next target instance, positioned around the previously found one. Although the size of this search region in pixels is partially adaptive, according to the target's image area on the previous video frame, it is practically limited by the video frame dimensions. Thus, the shot type requested by the director for a particular scenario at a certain time instance may not be feasible, depending on the specifics of the target and the camera motion velocities and trajectories.

Vertical Take-off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs, or "drones") equipped with professional cameras have recently become an indispensable asset in the cinematographer's arsenal. They permit rapid capture of impressive footage, flexible shot setup, novel shot types and access to narrow or hard-to-reach spaces, at a small fraction of the cost associated with spidercams, helicopters and cranes. Essentially, they provide a level of camera motion freedom that, so far, was only available in animation. Typically, in professional productions, the UAV and its mounted camera are manually remote-controlled by two different operators, acting in synchronization under a rough cinematography plan defined by the director. The latter can be conceived as a sequence of desired target assignments, shot types and UAV/camera motion trajectories relative to the target.

There is, however, a growing trend of increasing automation in drone functions, so as to reduce the challenges arising from fully manual operation [21] [24]. This is especially important in cinematography applications, where great precision and co-

ordination may be required in order to properly capture the desired shot. Thus, in the near future, production costs are expected to be significantly reduced, with semi-autonomous or fully autonomous drones replacing human crews currently required and shifting production focus to the direct realization of the director’s creative vision, rather than the minutiae of drone operation.

Autonomous UAV filming is, therefore, a promising emerging offshoot of intelligent shooting with potentially exceptional industrial impact. However, challenges such as tracking fast and unpredictably moving targets in real-time, as well as the lack of standardization in UAV shot types and meaningful UAV/camera motion trajectories, are a reality interfering with the ability to on-the-fly adjust the cinematography plan, according to dynamic environment conditions. The restrictions imposed on the feasible shot types by the requirements of the 2D visual tracker, especially, are particularly significant for autonomous UAVs, when contrasted with indoor robotic cameras, due to the possibly higher target speed in outdoor settings and the increased camera mobility offered by a drone.

Therefore, although the above apply to autonomous filming in general, this paper focuses on outdoor target-following UAV cinematography applications (e.g., for live sports event coverage). By significantly extending preliminary work [23] [40] [20] [22], it presents a theoretical study of the constraints imposed on cinematography decision-making during autonomous UAV shooting. The contributions of this paper are:

- Formalizing and geometrically modelling a range of common, target-following UAV motion types.
- Analytically determining the maximum permissible camera focal length f_{max} , so that 2D visual object tracking does not get lost, for each UAV motion type.
- Extracting formulas for determining the feasibility of the requested shot type (dependent on f_{max} and on the appropriate focal length f_s for that shot type).
- Providing specific examples and simulated scenarios that showcase the practical applicability of the proposed study.

Current industry practice simply ignores constraints implicitly imposed on zoom level/shot type by 2D visual tracker requirements. This is problematic, since it disregards the possibility of the target ROI going out of frame (or simply getting too spatially displaced in 2D pixel coordinates) among consecutive time instances, due to the target’s abrupt 3D motion and too high a focal length, thus breaking visual tracking. Therefore, to the best of our knowledge, our proposed, analytically derived rule set marks the first time this issue is studied in-depth in the context of autonomous UAV cinematography.

Incorporating shot type permissibility rules into media production automation software, such as intelligent UAV shooting algorithms [15] [16] [30] [35], is expected to greatly enhance the robustness of autonomous drones deployed in cinematography applications, by facilitating tracker-aware on-the-fly adjustment of the pre-computed cinematography plan.

Table 1: Shot types and their corresponding ROI to video frame height ratio percentage.

Shot type	Video frame height coverage
Extreme Long Shot (ELS)	< 5%
Very Long Shot (VLS)	5 – 20%
Long Shot (LS)	20 – 40%
Medium Shot (MS)	40 – 60%
Medium Close-Up (MCU)	60 – 75%
Close-Up (CU)	> 75%

2. UAV Cinematography Modelling

In cinematography, each camera motion type can be combined with a subset of the available shot types, so as to achieve an aesthetically pleasing visual result. Thus, a shot can be described by the combination of a camera motion type and a shot type. Below, shot types and camera motion types are studied for the specific case of UAV cinematography.

Each shot type is mainly defined by the ratio of the Region-of-Interest (ROI) height to the video frame height. The ratio can vary from less than 5% for the Extreme Long Shot, to more than 75% for Close-Up shot. The taxonomy presented in Table 1 is derived/adapted from traditional ground and aerial cinematography [5] [7] [34], based on extensive visual inspection of professional and semi-professional UAV footage.

In a typical scenario, the on-board camera is mounted on a *gimbal* that allows rapid camera rotation around its yaw, pitch and roll axes. Additionally, a zoom lens with adjustable focal length f (within certain limits) is employed. Simply altering f is typically sufficient for achieving the shot type desired by the director and prescribed in the cinematography plan. Thus, any constraints on the maximum permissible focal length directly correspond to restrictions in the range of feasible shot types at each time instance.

Regarding UAV/camera motion, several industry-standard types have emerged since the popularization of UAVs, with most of them being derived/adapted from traditional ground and aerial cinematography. For outdoor events (e.g., in live sports broadcasting), the most important motion types are relative to a still or moving target being tracked.

Recent aerial videography literature [7] [34] contains a description of a few such UAV motion types. However, no systematic analysis has been presented in the literature so far. Below, 8 UAV industry-standard camera motion types are detailed, geometrically modelled and matched to compatible shot types, based on our extensive visual survey of professional UAV footage. For instance, in a Chase shot (where the UAV follows/leads a moving target from behind/from the front, while maintaining a steady distance), the viewer is meant to experience a “simulation” of the target motion within its environment, while the target is fully visible. Thus, a CU that excludes most of the surroundings from the video frame is an unsuitable shot type in this context. Such findings are summarized in Table 2.

The mathematical treatment in this paper assumes a realistic setting similar to [35], where the autonomous UAV operates in a consistent, global, Cartesian 3D map, upon

Table 2: Compatibility of UAV camera motion and shot types.

Camera motion	Shot types
MAPMT	LS, MS, MCU
MATMT	LS, MS
LTS	VLS, LS, MS, MCU
VTS	VLS, LS, MS, MCU
ORBIT	LS, MS, MCU, CU
FLYOVER	LS, MS, MCU, CU
FLYBY	LS, MS, MCU, CU
CHASE	VLS, LS, MS

which both the drone itself and the target are constantly localized. This can be achieved by employing Global Positioning System (GPS) receivers [10] on both the UAV and the target. For increased robustness, GPS-derived drone localization information can be aligned and fused with Visual SLAM results [28], preferably derived by jointly exploiting stereoscopic 3D camera and Inertial Measurement Unit (IMU) [29] inputs, based on a similarity transformation [13]. Issues such as the possibility of temporarily losing the GPS signal, or the usual GPS position error (in the range of up to 5 meters [10]), may be overcome by fusing IMU/GPS and Visual SLAM localization, or by replacing GPS with an Active Radio-Frequency IDentification (RFID) positioning system [14]. Regarding the target, the output of 2D visual tracking itself can also be exploited for augmenting target localization precision (assuming a calibrated camera), thus making it even more imperative to reduce the chance of visual tracker failure.

Below, given a camera frame-rate F , time t is discrete and proceeds in steps of $\frac{1}{F}$ seconds. A separate timeline is employed for each shot description, i.e., $t = 0$ indicates the start of a shot shooting session. At each time instance t , the 3D positions $\tilde{\mathbf{x}}_t = [\tilde{x}_{t1}, \tilde{x}_{t2}, \tilde{x}_{t3}]^T$, $\tilde{\mathbf{p}}_t = [\tilde{p}_{t1}, \tilde{p}_{t2}, \tilde{p}_{t3}]^T$ of the UAV and the target respectively (assuming they are 3D points), as well as an estimated 3D target velocity vector $\tilde{\mathbf{u}}_t$, are assumed known (as in [35]) in a fixed, orthonormal, right-handed World Coordinate System (WCS), $\tilde{\mathbf{i}}, \tilde{\mathbf{j}}, \tilde{\mathbf{k}}$ with its $\tilde{\mathbf{k}}$ -axis perpendicular to a local tangent plane (hereafter shortened to “ground plane”). A local East-North-Up (ENU) coordinate system may be employed [9]. Note that the term “local tangent plane” is employed for a plane parallel to the local sea level, while the term “terrain tangent plane” is reserved for the plane instantaneously tangent to the local terrain surface.

Additionally, at each time instance t , a current, orthonormal, right-handed target-centered coordinate system (TCS), $\mathbf{i}, \mathbf{j}, \mathbf{k}$, is defined. Its origin lies on the current target position, its \mathbf{k} -axis is perpendicular to the ground plane and its \mathbf{i} -axis is the \mathcal{L}_2 -normalized projection of the current target velocity vector onto the ground plane. In the case of a still target, the TCS \mathbf{i} -axis is defined as parallel to the projection of the vector $\tilde{\mathbf{p}}_0 - \tilde{\mathbf{x}}_0$ onto the ground plane. In both coordinate systems, the \mathbf{ij} -plane is parallel to the ground plane and the \mathbf{k} -component is called “altitude”. Below, vectors expressed in TCS are denoted without the tilde symbol (e.g., \mathbf{x}_t , \mathbf{p}_t , \mathbf{q}_t and \mathbf{u}_t).

Transforming between the two coordinate systems is trivial. A subset of the presented motion types require pre-specification of motion parameters meant to adapt the

UAV motion trajectory to concrete directorial guidelines (e.g., distance to be covered by the UAV).

In mobile robotics literature, an additional, vehicle-centered coordinate system is typically employed, having its origin located at a fixed distance from the UAV-mounted camera. Since the scope of this paper does not include UAV control per se, we do not make use of such a coordinate frame and limit our analysis to cinematography issues. Additionally, for reasons of simplicity, the employed modelling ignores the distinction between the drone and its mounted camera, since it is typically trivial to compute the 3D pose of the one given the other and gimbal feedback.

The 3D scene point where the camera looks at time instance t , is denoted by \mathbf{l}_t (in TCS). The LookAt vector at time instance t is a scalar multiple of the camera axis and denoted by $\mathbf{o}_t = \mathbf{l}_t - \mathbf{x}_t$ (or $\tilde{\mathbf{o}}_t$, when expressed in WCS). Below, it is assumed that $\mathbf{l}_t = \mathbf{p}_t$ and, therefore, $\mathbf{o}_t = -\mathbf{x}_t$. As a result, the selected target point is visible at the center of the video frame. This is a simple and common framing approach, called “central composition”. Standard measurement units for the implicated quantities are also assumed, i.e., distance is measured in meters, speed in meters per second and the video frame-rate in frames per second.

In a number of cases, the UAV/camera motion type is only meaningful if the target is moving linearly. Moreover, such an assumption is additionally made below in cases where the future target or UAV position needs to be predicted, for reasons of modelling convenience (these cases are appropriately marked in the following analysis). Constant linear motion is assumed for both these scenarios, although extending the formulas for the case of constantly accelerated linear motion is trivial (assuming that the target acceleration vector can be reliably estimated).

The eight target-tracking UAV motion types are illustrated in Figure 1 and described below:

1) *Lateral Tracking Shot (LTS)* [7] [34] and 2) *Vertical Tracking Shot (VTS)* are non-parametric camera motion types, where the camera gimbal does not rotate and the camera is directly locked on the moving target. In LTS, the camera axis is approximately perpendicular both to the local target trajectory and to the WCS vertical axis vector \mathbf{k} , while the UAV flies sideways/in parallel to the target, matching its speed (if possible). In VTS, the camera axis is perpendicular to the target trajectory and the UAV flies exactly above the target, matching its speed (if possible). In both cases, $\tilde{\mathbf{p}}_t$ refers to a varying target position in WCS. During shooting, the UAV position remains constant in TCS, but varies in WCS.

The base mathematical description for both these UAV/camera motion types is fairly simple:

$$\tilde{\mathbf{v}}_t = \tilde{\mathbf{u}}_t, \quad \tilde{\mathbf{o}}_t^T \tilde{\mathbf{u}}_t \approx 0, \quad \mathbf{x}_t = \mathbf{x}_{t-1}, \quad \mathbf{l}_t = \mathbf{p}_t, \quad \forall t. \quad (1)$$

Additionally, the following relations hold for LTS and VTS, respectively:

$$\mathbf{o}_t \times \mathbf{j} \approx \mathbf{0}, x_{03} \approx 0, \quad (2)$$

$$\mathbf{o}_t^T \mathbf{j} \approx 0, x_{03} > 0. \quad (3)$$

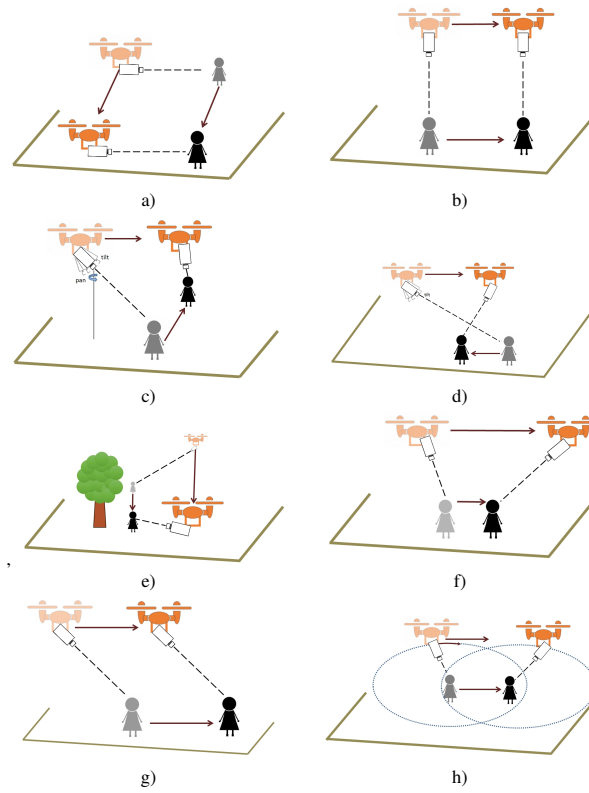


Figure 1: Examples of different target-tracking UAV camera motion types: a) Lateral Tracking Shot (LTS); b) Vertical Tracking Shot (VTS); c) Moving Aerial Pan with Moving Target (MAPMT); d) Moving Aerial Tilt with Moving Target (MATMT); e) Fly-By (FLYBY); f) Fly-Over (FLYOVER); g) Chase/Follow (CHASE); and h) Orbit (ORBIT) .

3) *Moving Aerial Pan with Moving Target (MAPMT)* and 4) *Moving Aerial Tilt with Moving Target (MATMT)* are parametric camera motion types, where the camera gimbal rotates (mainly with respect to the yaw/pitch axis, for MAPMT/MATMT, respectively) so as to always keep the linearly moving target centrally framed, while the UAV is flying at a linear trajectory with constant velocity. $\tilde{\mathbf{p}}_t$ refers to the target position, varying over time in such a manner that the target and the UAV velocity vector projections onto the ground plane are approximately perpendicular/parallel to each other, for MAPMT/MATMT, respectively.

The drone velocity vector $\tilde{\mathbf{v}}_t = [\tilde{v}_{t1}, \tilde{v}_{t2}, \tilde{v}_{t3}]^T$ must be specified. The base mathematical description for both these UAV/camera motion types is given by:

$$\tilde{\mathbf{v}}_t = \tilde{\mathbf{v}}_{t-1}, \quad \tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_0 + \frac{\tilde{\mathbf{v}}_t}{F}t, \quad \mathbf{l}_t = \mathbf{p}_t, \quad \forall t. \quad (4)$$

Additionally, the following relations hold for MAPMT and MATMT, respectively:

$$[\tilde{u}_{t1}, \tilde{u}_{t2}, 0][\tilde{v}_{t1}, \tilde{v}_{t2}, 0]^T \approx 0, \quad (5)$$

$$[\tilde{u}_{t1}, \tilde{u}_{t2}, 0]^T \times [\tilde{v}_{t1}, \tilde{v}_{t2}, 0]^T \approx \mathbf{0}. \quad (6)$$

5) *Fly-By (FLYBY)* and 6) *Fly-Over (FLYOVER)* [34]. They are parametric camera motion types, where the camera gimbal is rotating, so that the still or linearly moving target is always centrally framed. The UAV intercepts the target from behind/from the front (and to the left/right, in the case of FLYBY), at a steady altitude (in TCS) with constant velocity, flies exactly above it/passes it by (for FLYOVER/FLYBY, respectively) and keeps on flying at a linear trajectory, with the camera still pointing at the receding target. The UAV and target velocity vector projections onto the ground plane remain approximately parallel during shooting. They can have either identical or opposite direction. $\tilde{\mathbf{p}}_t$ refers to a varying or static target position in WCS.

The common parameter that must be specified is K , i.e., the time (in seconds) until UAV is located exactly above the target (for FLYOVER), or until the distance between the target and the UAV is minimized (for FLYBY). Additionally, the length d of the projection of that minimum distance vector onto the ground plane, must be specified for FLYBY. Below, the target velocity is assumed constant for reasons of modelling convenience. The mathematical description common to both camera motion types is the following one, for $t \in [0, 2KF]$:

$$\mathbf{v}_0 = \left[\frac{u_{01}K - x_{01}}{K}, 0, u_{03} \right]^T, \quad (7)$$

$$\tilde{\mathbf{v}}_t = \tilde{\mathbf{v}}_{t-1}, \quad \tilde{\mathbf{u}}_t = \tilde{\mathbf{u}}_{t-1}, \quad \mathbf{l}_t = \mathbf{p}_t, \quad \forall t, \quad (8)$$

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_0 + \frac{t}{KF}(\tilde{\mathbf{x}}_{KF} - \tilde{\mathbf{x}}_0), \quad (9)$$

$$[\tilde{u}_{t1}, \tilde{u}_{t2}, 0]^T \times [\tilde{v}_{t1}, \tilde{v}_{t2}, 0]^T \approx \mathbf{0}. \quad (10)$$

Additionally, the following relations holds for FLYOVER:

$$\tilde{\mathbf{x}}_{KF} = [\tilde{p}_{01} + \tilde{u}_{01}K, \tilde{p}_{02} + \tilde{u}_{02}K, \tilde{x}_{03} + \tilde{u}_{03}K]^T, \quad (11)$$

$$x_{t2} \approx 0, \quad \mathbf{x}_t^T \mathbf{j} \approx 0, \quad \forall t, \quad (12)$$

and the following hold for FLYBY:

$$|x_{02}| = d > 0, \quad x_{t2} = x_{02}, \quad \forall t, \quad (13)$$

$$\mathbf{x}_{KF} = [0, x_{02}, x_{03}]^T. \quad (14)$$

7) *Chase/Follow Shot (CHASE)* is a non-parametric camera motion type, where the camera gimbal does not rotate and the camera always points at the target [34]. The UAV follows/leads the target from behind/from the front, while maintaining a steady distance by matching its speed, if possible. $\tilde{\mathbf{p}}_t$ refers to a varying target position in WCS. The mathematical description is the following:

$$\tilde{\mathbf{v}}_t \approx \tilde{\mathbf{u}}_t, \quad (15)$$

$$x_{t2} = x_{02} \approx 0, \quad \mathbf{x}_t = \mathbf{x}_{t-1}, \quad \mathbf{l}_t = \mathbf{p}_t, \quad \forall t. \quad (16)$$

8) *Orbit (ORBIT)*. It is a parametric camera motion type, where the camera gimbal is slowly rotating, so as to always keep the still or linearly moving target properly framed, while the UAV (semi-)circles around the target and, simultaneously, follows the target linear trajectory (if the target is moving) [7] [34]. During shooting, the UAV altitude remains constant in TCS, but may vary in WCS. $\tilde{\mathbf{p}}_t$ refers to a varying or static target position in WCS.

The parameters that must be specified are the desired 3D Euclidean distance $d_{3D} = \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{p}}_t\|_2 = \|\mathbf{x}_t\|_2$ (constant over time), the rotation angle θ around the target and the desired UAV angular velocity ω . Additionally, we can easily derive the initial angle θ_0 formed by the TCS \mathbf{i} -axis (of time instance $t = 0$) and the vector from \mathbf{p}_0 to the projection of the known initial position \mathbf{x}_0 onto the TCS \mathbf{ij} -plane. Then, ORBIT may be described in TCS using a planar circular motion, for $t \in [0, \frac{T\theta}{\omega}]$:

$$\theta_0 = \arctan\left(\frac{x_{02}}{x_{01}}\right), \quad (17)$$

$$x_{t3} = x_{03}, \forall t, \quad (18)$$

$$\lambda = \sqrt{\lambda_{3D}^2 - x_{t3}^2}, \quad (19)$$

$$\mathbf{x}_t = [\lambda \cos(t\frac{\omega}{F} + \theta_0), \lambda \sin(t\frac{\omega}{F} + \theta_0), x_{t3}]^T, \quad (20)$$

$$\mathbf{l}_t = \mathbf{p}_t. \quad (21)$$

3. Constraints on Maximum Focal Length

In order for a visual tracker to operate properly, the location (in pixel coordinates) of the target ROI should differ no more than a threshold between successive video frames/time instances. This requirement places a constraint on the maximum target speed and on the maximum camera focal length f (the main factor determining maximum achievable zoom level), since a given 3D target displacement (in WCS) corresponds to a greater 2D ROI displacement (in pixels) at a greater zoom level. Proper estimation of the maximum allowable f in each shooting case is of utmost importance in cinematography applications, since it directly affects the range of permissible shot types.

Without loss of generality, we always consider time instance $t = 0$ and, thus, examine an entire shooting session as a sequence of repeated transitions between the “first” ($t = 0$) and the “second” video frame ($t + 1 = 1$). We also assume that the target ROI center is always meant to be fixed at the principal point (image center) of all video frames (central composition). Target position $\tilde{\mathbf{p}}_t$ is initially known and $\tilde{\mathbf{p}}_{t+1}$ can be predicted using the estimated velocity vector $\tilde{\mathbf{u}}_t$, i.e., $\tilde{\mathbf{p}}_{t+1} = \tilde{\mathbf{p}}_t + \tilde{\mathbf{u}}_t \frac{1}{F}$. If the prediction is accurate, the target ROI indeed remains at the center of the $(t + 1)$ -th video frame.

In contrast, if the actual current target motion differs from the predicted one by the unknown velocity deviation vector $\tilde{\mathbf{q}}_t = [\tilde{q}_{t1}, \tilde{q}_{t2}, \tilde{q}_{t3}]^T$, the target ROI at time $t + 1$ has to be explicitly localized via 2D visual tracking (in pixel coordinates), so that it can be exploited for 3D target position $\tilde{\mathbf{p}}_{t+1}$ estimation and/or for adjusting the framing. Since $\tilde{\mathbf{q}}_t$ and, therefore, $\tilde{\mathbf{p}}_{t+1}$ are unknown, the following analysis utilizes the TCS defined by the expected/predicted target position at time instance $t + 1$.

Whenever $\tilde{\mathbf{q}}_t$ is a non-zero vector and, therefore, prediction of $\tilde{\mathbf{p}}_{t+1}$ fails, the results of 2D visual tracking and actual $\tilde{\mathbf{p}}_{t+1}$ estimation must be employed for updating the target velocity vector and, hopefully, achieving a better prediction during the next time instance. Given that tracker behavior varies per algorithm, we simply assume a maximum search radius R_{max} (in pixels) defining the video frame region within which the tracked object ROI of time instance $t + 1$ must lie, relatively to the video frame center, in order to permit successful tracking. Thus, a distance R_{t+1} between the actual target ROI center of $t + 1$ and the center of that video frame, where $R_{t+1} > R_{max}$, implies tracking failure. The case where $R_{t+1} = R_{max}$ marks the limit scenario where the tracker marginally succeeds. Note that R_{max} is not fixed, since modern trackers adapt the size of their search region to the current ROI size.

3.1. Maximum focal length

In order to find the maximum focal length so that there is no target tracking failure, we assume that the expected position of the target in TCS is always at $[0, 0, 0]^T$. Let $\mathbf{o}_t = \mathbf{l}_t - \mathbf{x}_t$ be the LookAt vector at time instance t and $d_t = \sqrt{x_{t1}^2 + x_{t2}^2}$ is the distance between the target and the UAV, projected on the \mathbf{ij} -plane.

Based on the above and the camera projection equations [36], the following hold:

$$x_d(t + 1) = o_x - \frac{f}{s_x} \frac{\mathbf{r}_1^T (\mathbf{p}_{t+1} - \mathbf{x}_{t+1})}{\mathbf{r}_3^T (\mathbf{p}_{t+1} - \mathbf{x}_{t+1})}, \quad (22)$$

$$y_d(t+1) = o_y - \frac{f}{s_y} \frac{\mathbf{r}_2^T (\mathbf{p}_{t+1} - \mathbf{x}_{t+1})}{\mathbf{r}_3^T (\mathbf{p}_{t+1} - \mathbf{x}_{t+1})}, \quad (23)$$

where $x_d(t+1)$, $y_d(t+1)$ are the target center pixel coordinates at the time instance $(t+1)$, o_x , o_y define the image center in pixel coordinates and s_x , s_y denote the pixel size (in mm) along the horizontal and vertical directions. \mathbf{r}_1 , \mathbf{r}_2 and \mathbf{r}_3 refer, respectively, to the first, second and third row of the rotation matrix \mathbf{R} that orients the camera gimbal according to the LookAt vector.

In general, the coordinate transform matrix from TCS to the camera coordinate system can be found by two rotations and one translation of the unit TCS vectors. The required rotations are around the TCS \mathbf{k} -axis and \mathbf{j} -axis. Thus, \mathbf{R} can be described as follows [2]:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta_z)\cos(\theta_y) & -\sin(\theta_z) & \cos(\theta_z)\sin(\theta_y) \\ \sin(\theta_z)\cos(\theta_y) & \cos(\theta_z) & \sin(\theta_z)\sin(\theta_y) \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix}, \quad (24)$$

where θ_z and θ_y are the appropriate angles of rotation for \mathbf{R}_z and \mathbf{R}_y respectively. However, given that \mathbf{R} is an orthogonal change-of-basis matrix and that, in most of the motion types, the UAV does not fly exactly above the target, it is easier to obtain the rows of \mathbf{R} as follows. Since the camera axis points directly at the target, the unit vector of the \mathbf{k} -axis for the Camera Coordinate System, i.e., \mathbf{r}_3 , can be obtained from \mathbf{x}_{t+1} as follows:

$$\mathbf{r}_3 = \left(\frac{-\mathbf{x}_{t+1}}{\|\mathbf{x}_{t+1}\|} \right)^T. \quad (25)$$

For motion types where the UAV does not fly exactly above the target, \mathbf{r}_1 is the cross product of \mathbf{r}_3 with the unit vector \mathbf{k} :

$$\mathbf{r}'_1 = \left(\mathbf{k} \times \frac{-\mathbf{x}_{t+1}}{\|\mathbf{x}_{t+1}\|} \right)^T, \quad (26)$$

$$\mathbf{r}_1 = \frac{\mathbf{r}'_1}{\|\mathbf{r}'_1\|}. \quad (27)$$

Thus, \mathbf{r}_2 is given by the cross product $\mathbf{r}_3 \times \mathbf{r}_1$:

$$\mathbf{r}'_2 = \left(\frac{-\mathbf{x}_{t+1}}{\|\mathbf{x}_{t+1}\|} \times \left(\mathbf{k} \times \frac{-\mathbf{x}_{t+1}}{\|\mathbf{x}_{t+1}\|} \right) \right)^T, \quad (28)$$

$$\mathbf{r}_2 = \frac{\mathbf{r}'_2}{\|\mathbf{r}'_2\|}. \quad (29)$$

In our approach we consider central composition, thus the target ROI center should be located at (o_x, o_y) at all times. Assuming that in time instance t the target ROI center is aligned with the frame center, in time instance $t' = t + 1$, the target ROI center will be translated to a new pixel coordinates, due to camera and target movement in the real world. The central pixel translation of the ROI, R , can be calculated by employing

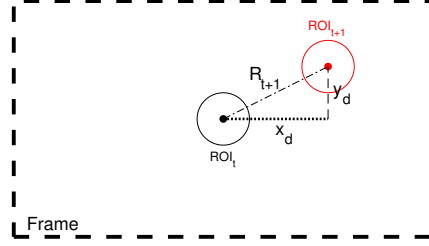


Figure 2: ROI translation between two consecutive video frames for time instance t and $t' = t + 1$. The distance between the central pixels of the two ROIs, R can be calculated by employing the results of Eqs. (22) and (23).

Eqs. (22) and (23), and simple geometrical rules, as depicted in Fig. 2. By setting a maximum R value, thus applying the limit constraint $R_{t+1} = R_{max}$, we derive the following equation:

$$R_{max} = \sqrt{(x_d(t+1) - o_x)^2 + (y_d(t+1) - o_y)^2}. \quad (30)$$

Assuming that $\mathbf{x}_{t'} = [x_{t'1}, x_{t'2}, x_{t'3}]^T$ and $\mathbf{p}_{t'} = [\frac{q_{t1}}{F}, \frac{q_{t2}}{F}, \frac{q_{t3}}{F}]^T$, where $t' = t + 1$, and substituting Eqs. (22) and (23) in Eq. (30), R_{max} can be obtained by:

$$R_{max} = \sqrt{f_{max}^2 \|\mathbf{x}_{t'}\|^2 \left(\frac{E_3^2}{s_x^2} + \frac{(q_{t3}N - E_2x_{t'3})^2}{s_y^2(N + x_{t'3}^2)} \right)} \quad (31)$$

where

$$N = (x_{t'1}^2 + x_{t'2}^2)$$

Eq. (31) can be solved for f to obtain the maximum focal length f_{max} for motion types having $d_{t'} > 0$:

$$f_{max} = \frac{R_{max}d_{t'}s_xs_y|E_1 + F\|\mathbf{x}_{t'}\|^2|}{\sqrt{(s_xq_{t3}d_{t'}^2 - s_xx_{t'3}E_2)^2 + s_y^2E_3^2\|\mathbf{x}_{t'}\|^2}}, \quad (32)$$

where

$$E_1 = -q_{t1}x_{t'1} - q_{t2}x_{t'2} - q_{t3}x_{t'3},$$

$$E_2 = q_{t1}x_{t'1} + q_{t2}x_{t'2},$$

$$E_3 = q_{t2}x_{t'1} - q_{t1}x_{t'2}.$$

Since most of the UAV motion types are not affected by target altitude changes between successive video frames, which are less likely to happen than direction and

speed changes, $\mathbf{p}_{t'}$ can be expressed as follows:

$$\mathbf{p}_{t'} = [\frac{q_{t1}}{F}, \frac{q_{t2}}{F}, 0]^T. \quad (33)$$

In this case, the maximum focal length is given by:

$$f_{max} = \frac{R_{max}d_{t'}s_xs_y| - E_2 + F \|\mathbf{x}_{t'}\|^2 |}{\sqrt{s_x^2E_2^2x_{t'3}^2 + s_y^2E_3^2\|\mathbf{x}_{t'}\|^2}}. \quad (34)$$

When the UAV/camera is located exactly above the target for the $(t + 1)$ -th video frame, i.e., $\mathbf{x}_{t'} = [0, 0, x_{t'3}]^T$, \mathbf{R} cannot be derived as described in Eqs. (25)-(29), since $\mathbf{r}_1 \times \mathbf{k} = \mathbf{0}$. In this special case, where $d_{t'} = 0$, it is easier to calculate the rotation matrix using (24), for $\theta_z = 0$ and $\theta_y = 180^\circ$:

$$\mathbf{R} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (35)$$

Then, the maximum focal length is given by:

$$f_{max} = \frac{R_{max}Fx_{t'3}s_xs_y}{\sqrt{s_y^2q_{t1}^2 + s_x^2q_{t2}^2}}. \quad (36)$$

As it can be seen from the above, in general, the derived formulas rely on knowing, predicting or estimating a velocity deviation vector \mathbf{q}_t that models the degree to which instantaneous target 3D motion differs from uniform linear motion. Several options are available for obtaining \mathbf{q}_t . A reasonable choice would be to assume an instantaneously constant acceleration vector at each time instance. A more strict policy would be to derive f_{max} for various candidate velocity deviations, which displace the target towards different spatial directions, and output the minimum among the computed f_{max} values.

3.2. Simulations for specific UAV/camera motion types

In order to investigate the maximum possible focal length for a specific motion type shot, we simulated the motion for various representative UAV shooting scenarios. We studied 8 different cases for the deviation vector \mathbf{q}_t . In the first two cases, the target linearly accelerates/decelerates, i.e., $\mathbf{q}_{t1} = [7.5, 0, 0]^T$, $\mathbf{q}_{t2} = [-7.5, 0, 0]^T$. Velocity deviations are expressed in meters/second. In the third and fourth cases, the target is moving along a different direction than the expected one ($\mathbf{q}_{t3} = [0, 7.5, 0]^T$, $\mathbf{q}_{t4} = [0, -7.5, 0]^T$), but remains on the TCS \mathbf{j} -axis. In the remaining cases, the target is moving diagonally to the TCS axes ($\mathbf{q}_{t5} = [7.5, 7.5, 0]^T$, $\mathbf{q}_{t6} = [-7.5, -7.5, 0]^T$, $\mathbf{q}_{t7} = [-7.5, 7.5, 0]^T$, $\mathbf{q}_{t8} = [7.5, -7.5, 0]^T$). Figure 3 depicts the expected against the actual position of the target in each case.

The following parameters have been used in the performed simulations. Maximum tracker search radius R_{max} was generously fixed to 360 pixels, so as to model the obvious constraint that the central target ROI pixel stays visible among consecutive video frames (when using High Definition camera sensor), otherwise visual tracking

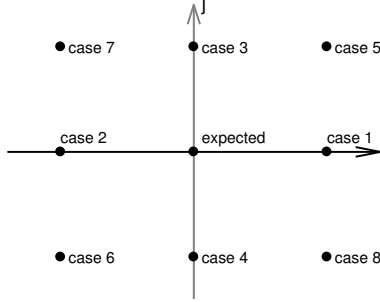


Figure 3: The expected against the actual target position in the $(t + 1)$ -th time instance, for the 8 simulated cases. TCS i and j axes are denoted by black and grey color, respectively.

fails. This is a hard upper bound on R_{max} , thus bypassing the need for adaptive R_{max} in this set of experiments. The pixel size was set to $s_x = s_y = 0.009$ mm and video frame rate to $F = 25$ fps. All of the experiments were carried out on a Linux PC equipped with an Intel i7 CPU and 32 GB of RAM. However, the proposed rules can be easily computed in real-time on an embedded system (e.g. nVidia Jetson, Intel NUC, etc.), in conjunction with a fast 2D visual tracker.

3.2.1. Lateral Tracking Shot

In LTS, the UAV flies alongside the target, as described in Section 2. In this shot type, even small target altitude variations have a great impact on picture framing. Therefore, we assume that $q_{t3} \neq 0$. The UAV position is given by $\mathbf{x}_{t+1} = [0, x_{t2}, 0]^T$. As $\mathbf{p}_{t+1} = [\frac{q_{t1}}{F}, \frac{q_{t2}}{F}, \frac{q_{t3}}{F}]^T$, Eq. (32) can now be rewritten as follows:

$$f_{max} = \frac{R_{max} s_x s_y |q_{t2} - F x_{t2}|}{\sqrt{s_y^2 q_{t1}^2 + s_x^2 q_{t3}^2}}. \quad (37)$$

The LTS simulation was performed for varying values of q_{t3} . The horizontal distance between the UAV and the target was chosen to be $\lambda = x_{t2} = 30m$. Simulation results are shown in Figure 4. As expected, variations in altitude affect all study cases 1 - 8. When the target deviates from its expected TCS position $[0, 0, 0]^T$, but is located on the j -axis, i.e., $\mathbf{p}_{t+1} = [0, \frac{q_{t2}}{F}, 0]^T$, f_{max} is only affected by altitude changes. This behavior is reasonable, since the camera k -axis unit vector can be expressed in TCS as $\mathbf{k}_c = [0, -1, 0]^T$. Consequently, the projected ROI center will not change in pixel coordinates, therefore, this target deviation should have no impact at all on f_{max} , when $q_{t3} = 0$. The other results are affected by linear target acceleration/deceleration along the TCS i -axis. As expected, f_{max} is maximized for these cases (1, 2 and 5 - 8) when the target altitude does not vary between successive video frames. Due to the position of the UAV, target acceleration and deceleration have identical impact on f_{max} .

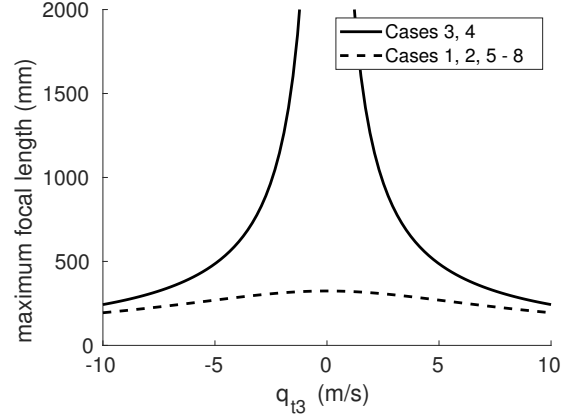


Figure 4: Simulation results for LTS: f_{max} against q_{t3} .

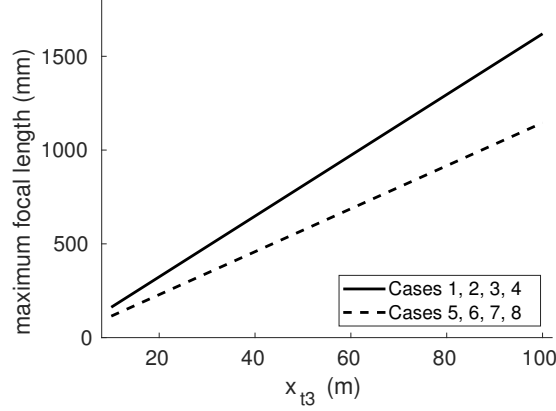


Figure 5: Simulation results for VTS: f_{max} against altitude (x_{t3}).

3.2.2. Vertical Tracking Shot

In VTS, the UAV flies exactly above the target, therefore, the maximum focal length is given by Eq. (36). The UAV is positioned at $\mathbf{x}_{t+1} = [0, 0, x_{t3}]^T$. The 8 case studies were simulated for various UAV TCS altitudes, i.e., for various values of x_{t3} . Thus, we obtained the maximum focal length allowed in the VTS scenario for various UAV altitudes, under the assumption that target altitude remains approximately constant between successive video frames, i.e., $q_{t3} = 0$. Target position at time $t + 1$ is given by: $\mathbf{p}_{t+1} = [\frac{q_{t1}}{F}, \frac{q_{t2}}{F}, 0]^T$. The results are presented in Figure 5, where the horizontal axis unit is meters and the vertical axis unit is millimetres. As expected, the maximum focal length increases linearly with x_{t3} . When the target is moving diagonally to the TCS axes (cases 5 - 8) the maximum possible focal length is lower than in cases 1 - 4. Target motion along the j -axis (cases 3 and 4) and target linear acceleration (cases 1 and 2) have similar effect on the maximum allowed focal length, since the UAV is positioned exactly above the target.

3.2.3. Moving Aerial Pan with Moving Target/Moving Aerial Tilt with Moving Target

Given the mathematical description for MAPMT/MATMT in (4) and the fact that the target is moving along the *i*-axis, we can assume that $\mathbf{x}_{t+1} = [x_{t1}, x_{t2} + \frac{v_{t2}}{F}, x_{t3}]^T$ for MAPMT and $\mathbf{x}_{t+1} = [x_{t1} + \frac{v_{t1}}{F}, x_{t2}, x_{t3}]^T$ for MATMT. For the UAV position at time instance $t + 1$, the target position in the next video frame is given by Eq. (33). By substituting \mathbf{x}_{t+1} in Eq. (34), the following relations hold:

$$f_{max} = \frac{R_{max}d_{mp}s_xs_y| - E_{mp1} + F \|\mathbf{x}_{t+1}\|^2 |}{\sqrt{s_x^2E_{mp1}^2x_{t3}^2 + s_y^2E_{mp2}^2 \|\mathbf{x}_{t+1}\|^2}} \quad (38)$$

$$f_{max} = \frac{R_{max}d_{mt}s_xs_y| - E_{mt1} + F \|\mathbf{x}_{t+1}\|^2 |}{\sqrt{s_x^2E_{mt1}^2x_{t3}^2 + s_y^2E_{mt2}^2 \|\mathbf{x}_{t+1}\|^2}} \quad (39)$$

for MAPMT and MATMT, respectively, where:

$$\begin{aligned} d_{mp} &= \sqrt{x_{t1}^2 + (x_{t2} + \frac{v_{t2}}{F})^2}, \\ E_{mp1} &= q_{t1}x_{t1} + q_{t2}(x_{t2} + \frac{v_{t2}}{F}), \\ E_{mp2} &= q_{t2}x_{t1} + q_{t1}(x_{t2} + \frac{v_{t2}}{F}), \\ d_{mt} &= \sqrt{x_{t2}^2 + (x_{t1} + \frac{v_{t1}}{F})^2}, \\ E_{mt1} &= q_{t2}x_{t2} + q_{t1}(x_{t1} + \frac{v_{t1}}{F}), \\ E_{mt2} &= q_{t1}x_{t2} + q_{t2}(x_{t1} + \frac{v_{t1}}{F}). \end{aligned}$$

For simulation purposes, f_{max} was studied for varying distances between the target and the UAV, corresponding to consecutive time instances of the UAV/camera motion type execution. The following initial values were selected: $x_{01} = 30 \text{ m}$, $x_{02} = -60 \text{ m}$ (MAPMT), $x_{01} = -60 \text{ m}$, $x_{02} = 30 \text{ m}$ (MATMT), $x_{03} = 10 \text{ m}$, $v_{t2} = 10 \frac{\text{m}}{\text{s}}$ (both). The similarities between Figures 6 and 7, for MAPMT and MATMT, respectively, are evident. As expected, cases 1, 2/3, 4 of MAPMT correspond to cases 3, 4/1, 2 of MATMT, since these two motion types differ only in the UAV motion direction: it is parallel to the *j*-axis/*i*-axis in MAPMT/MATMT, respectively. The impact on f_{max} for target motion deviation along the TCS *j*-axis for MAPMT will be the same as the impact for target motion deviation along the TCS *i*-axis for MATMT, and vice versa, as Figure 8 demonstrates. Therefore, cases 5, 6 and 7, 8 produce identical results in both motion types.

Studying the results of cases 1 and 2 for MAPMT and cases 3 and 4 for MATMT, f_{max} takes its maximum value when $x_{t2} = 0$ and $x_{t1} = 0$, respectively. The reason is that, in these positions, the UAV in MAPMT is above the *i*-axis, while in MATMT above the *j*-axis, thus any deviations in target motion affect minimally the ROI location in the next video frame. On the other hand, in all other cases, these UAV positions are approximately where any target motion deviations have the greatest impact on the next ROI location.

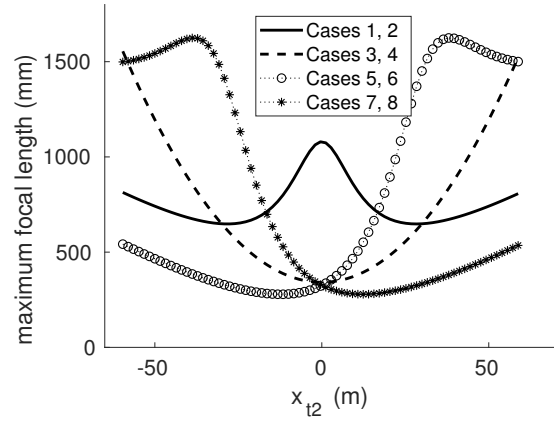


Figure 6: Simulation results for MAPMT: f_{max} against x_{t2} .

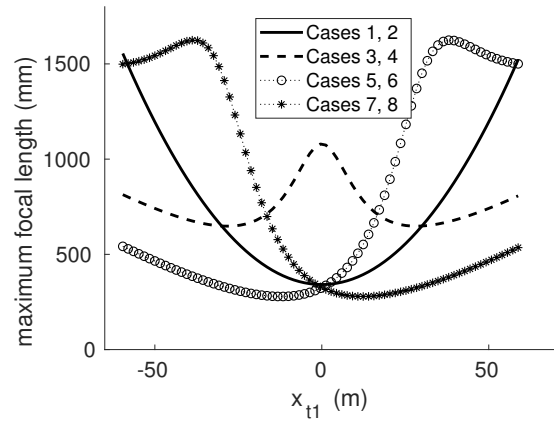


Figure 7: Simulation results for MATMT: f_{max} against x_{t1} .

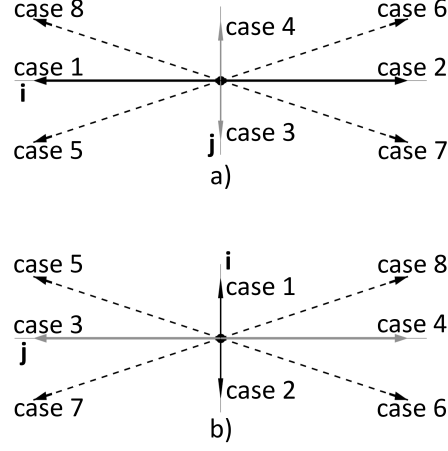


Figure 8: Target velocity deviation vectors as seen from the UAV camera, when the camera axis lies on: a) the \mathbf{j} -axis and b) the \mathbf{i} -axis. Black dot denotes target expected position. Black vectors correspond to cases 1 and 2, grey vectors to cases 3 and 4 and, finally, the dashed lined vectors to cases 5-8. In a) target velocity deviation on the \mathbf{j} -axis will affect less the f_{max} than target linear speed changes, while in b) the opposite.

3.2.4. Fly-By/Fly-Over

In these motion types, where shot duration is specified by K , we can determine the maximum focal length directly over time ($t \in [0, 2K]$). For FLYBY, the UAV position in TCS is given by $\mathbf{x}_{t+1} = [-\frac{x_{01}}{K}t + x_{01}, x_{02}, x_{03}]^T$. We study these motion types together, since FLYOVER is a special case of FLYBY, where $x_{02} = 0$.

By substituting x_{t+1} in Eq. (34), f_{max} is given by:

$$f_{max} = \frac{R_{max}d_{fb}s_x s_y | -E_{fb1} + F \| \mathbf{x}_{t+1} \|^2 |}{\sqrt{s_x^2 E_{fb1}^2 x_{t3}^2 + s_y^2 E_{fb2}^2 \| \mathbf{x}_{t+1} \|^2}}, \quad (40)$$

$$f_{max} = \frac{R_{max}d_{fo1}s_x s_y | -E_{fo1} + F \| \mathbf{x}_{t+1} \|^2 |}{\sqrt{s_x^2 E_{fo1}^2 x_{t3}^2 + s_y^2 E_{fo2}^2 \| \mathbf{x}_{t+1} \|^2}}, \quad (41)$$

for FLYBY and FLYOVER, respectively, where:

$$\begin{aligned}
d_{fb} &= \sqrt{\left(\frac{-x_{01}}{K}t + x_{01}\right)^2 + x_{02}^2}, \\
E_{fb1} &= q_{t1}\left(\frac{-x_{01}}{K}t + x_{01}\right) + q_{t2}x_{t2}, \\
E_{fb2} &= q_{t2}\left(\frac{-x_{01}}{K}t + x_{01}\right) - q_{t1}x_{t2}, \\
d_{fo} &= \left|\left(\frac{-x_{01}}{K}t + x_{01}\right)\right|, \\
E_{fo1} &= q_{t1}\left(\frac{-x_{01}}{K}t + x_{01}\right), \\
E_{fo2} &= \left(q_{t2}\left(\frac{-x_{01}}{K}t + x_{01}\right)\right).
\end{aligned}$$

The following parameter values were chosen for the simulation: $x_{01} = -30 \text{ m}$, $x_{03} = 10 \text{ m}$, $K = 10$, thus $t \in [0, 20]$. Additionally, $x_{02} = 15 \text{ m}$ for FLYBY. Results are shown in Figures 9 and 10, for FLYBY and FLYOVER, respectively. The gap in FLYOVER for $t = 10$ stems from the fact that the UAV is actually above the target and, thus, the motion type is momentarily converted to VTS.

In cases 1 and 2, both motion types produce similar results. As the UAV approaches the target, the maximum focal length decreases, before increasing again as the UAV is flying parallel to the i -axis. When the drone is positioned far from the target, any change in target speed corresponds to a small change in the distance between the UAV and the target.

In general, for cases 3 and 4 of FLYBY, where the target deviates from its expected position but remains on the j -axis, f_{max} increases with rising distance between the UAV and the target. Additionally, f_{max} also slightly increases when the UAV is very close to the target. Then, the latter's velocity deviation corresponds to a small change in distance between the target and the UAV, mapped to a small ROI displacement and, thus, greater focal length tolerance. In FLYOVER, where any deviation of the target motion on the j -axis will always displace the target ROI to the left or right of the video frame, f_{max} is significantly smaller for cases 3 and 4.

Finally, in cases 5-8 of FLYBY, f_{max} depends on the angle between the LookAt vector and the i -axis: it has lower values when this angle is close to $\frac{\pi}{2}$ ($t = 10$ in the simulation). In FLYOVER, the overall minimum values of f_{max} are also obtained for cases 5-8 when $t = 10$, since, then, the 3D distance between the expected and the actual target position is slightly greater compared to cases 1-4, as it can be seen in Figure 3, leading to greater 2D ROI displacement.

3.2.5. Chase

The focal length constraint for this motion type is a special case of Eq. (34) where $x_{t2} = 0$. Since the UAV is always located in front of/behind the target and at a steady distance, its position at time instance $t + 1$ is given by $\mathbf{x}_{t+1} = [x_{t1}, 0, x_{t3}]^T$. Target position in the next time instance is given by Eq. (33). By combining (33) and (34),

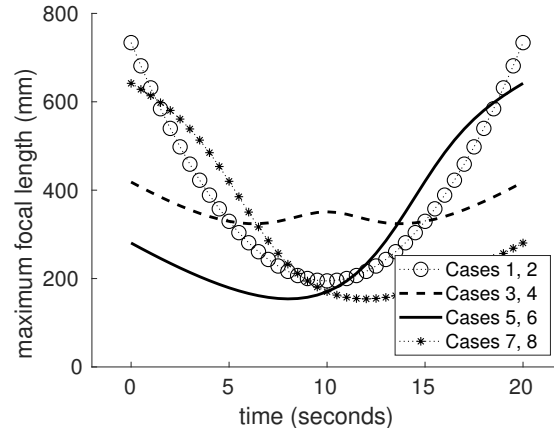


Figure 9: Simulation results for FLYBY: f_{max} over time t .

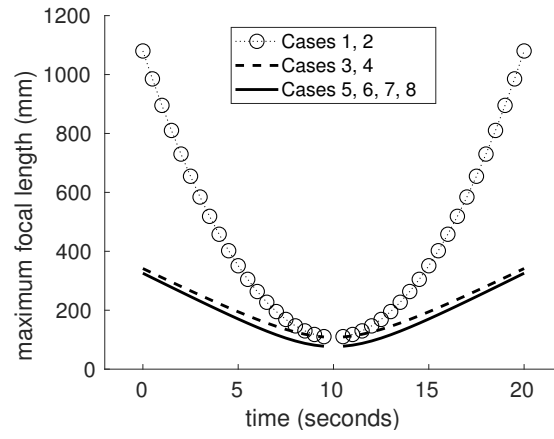


Figure 10: Simulation results for FLYOVER: f_{max} over time t .

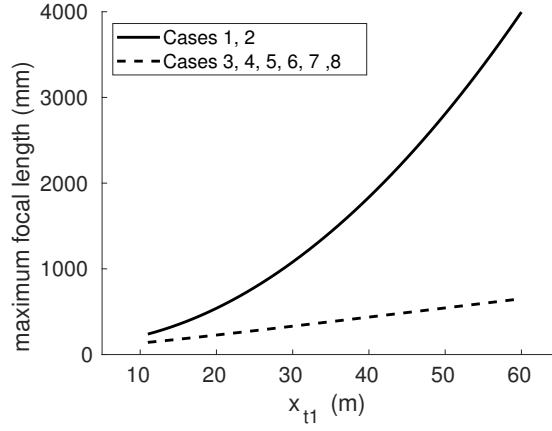


Figure 11: Simulation results for CHASE: f_{max} against distance from target.

the following relation holds:

$$f_{max} = \frac{R_{max} s_x s_y \phi_c | - F \phi_c^2 + x_{t1} q_{t1} |}{x_{t1} \sqrt{s_y^2 \phi_c^2 q_{t2}^2 + s_x^2 x_{t3}^2 q_{t1}^2}}, \quad (42)$$

where

$$\phi_c = \sqrt{x_{t1}^2 + x_{t3}^2}. \quad (43)$$

For simulation purposes, we studied f_{max} using varying distances between the target and the UAV, as well as constant TCS altitude ($x_{t3} = 10 \text{ m}$). The results are shown in Figure 11. As expected, the maximum focal length increases with rising distance between the UAV and the target. In cases 1 and 2, f_{max} is much larger than in the other cases, since an increase or a decrease of the target speed will simply move the target slightly away or closer to the UAV. When distance between the UAV and the target is increased, the target has to deviate more from its expected position, so that $R_{t+1} > R_{max}$ in the next video frame. This is due to the fact that target speed deviation has less effect on target position in the next video frame, as this UAV/camera motion type starts to produce a visual result similar to that of LTS, but with the UAV located ahead/behind the target.

On the contrary, for cases 3 and 4 where the target deviates along the j -axis in the next video frame, this UAV/camera motion type is highly affected. As Figure 8b demonstrates, if the target moves along the j -axis, the ROI center in the next video frame is displaced according to target motion velocity deviation. However, this displacement is also inversely proportional to the distance between the target and the UAV/camera, due to perspective projection. Thus, lower focal length tolerances and a more linear increase in f_{max} as x_{t1} rises is expected. Similar conclusions can be drawn for cases 5 - 8.

3.2.6. Orbit

For the ORBIT motion type, the target position is given by Eq. (33). By using Eqs. (17) - (21), f_{max} is given by substituting

$$\mathbf{x}_{t+1} = [\lambda \cos(\frac{\omega}{F} + \theta_0), \lambda \sin(\frac{\omega}{F} + \theta_0), x_{t3}]^T \quad (44)$$

in (34):

$$f_{max} = \frac{R_{max} d_{or} s_x s_y | - E_{or1} + F \| \mathbf{x}_{t+1} \|^2 |}{\sqrt{s_x^2 E_{or1}^2 x_{t3}^2 + s_y^2 E_{or2}^2 \| \mathbf{x}_{t+1} \|^2}}, \quad (45)$$

where:

$$\begin{aligned} d_{or} &= \sqrt{(\lambda \cos(\frac{\omega}{F} + \theta_0))^2 + (\lambda \sin(\frac{\omega}{F} + \theta_0))^2}, \\ E_{or1} &= q_{t1} \lambda \cos(\frac{\omega}{F} + \theta_0) + q_{t2} \lambda \sin(\frac{\omega}{F} + \theta_0), \\ E_{or2} &= q_{t1} \lambda \sin(\frac{\omega}{F} + \theta_0) + q_{t2} \lambda \cos(\frac{\omega}{F} + \theta_0). \end{aligned}$$

The following parameter values were used in the simulations: $\lambda = 30$ m, $x_{03} = 10$ m, $\omega = \frac{\pi}{20}$ rad/sec. The results are depicted in Figure 12. The horizontal axis represents the current θ_0 , i.e., the angle denoting the current UAV position relative to the target along a circular trajectory. The estimated f_{max} complies with intuitive expectations in all cases. For instance, in case 1, the target linearly accelerates. If the UAV lies exactly behind the target ($\theta_0 = 0^\circ$), f_{max} takes its maximum value, since, from that perspective, a linear acceleration will not significantly alter the target ROI center pixel coordinates. In contrast, linear acceleration will have a much greater impact from a lateral perspective ($\theta_0 = 90^\circ$). Indeed, f_{max} takes its minimum value in this case. As expected, f_{max} varies periodically as the UAV view changes from a lateral one to a collinear one and vice versa. Similar conclusions can be drawn for the scenario of linear target deceleration (case 2), where the target trajectory also remains identical to the expected one.

In cases 3 and 4, if the UAV is positioned collinearly to the estimated target velocity vector ($\theta_0 = 0^\circ$), it has in fact a lateral view of the actual target motion. If it is positioned perpendicularly to the estimated velocity vector ($\theta_0 = 90^\circ$), it has in fact a collinear (frontal/rear) view of the actual target motion. Therefore, the plots of the cases 1, 2 and of the cases 3, 4 have a relative phase difference of $\frac{\pi}{2}$, as one would expect.

As shown in Figure 12, in cases 5 and 6, where the target moves diagonally to its expected trajectory, the corresponding plots have an absolute phase difference of $\frac{\pi}{8}$ relative to the previously described plots. Additionally, the f_{max} values are lower than those of cases 3 and 4. These observations are reasonable, since, when $\theta_0 = 45^\circ$, the UAV has in fact a frontal/rear view of the actual target motion. Also, this scenario presents the greatest difference (in pixel coordinates) between the expected and the actual target ROI center location. Therefore, greater limitations are naturally imposed on f_{max} , so that 2D visual tracking is successful.

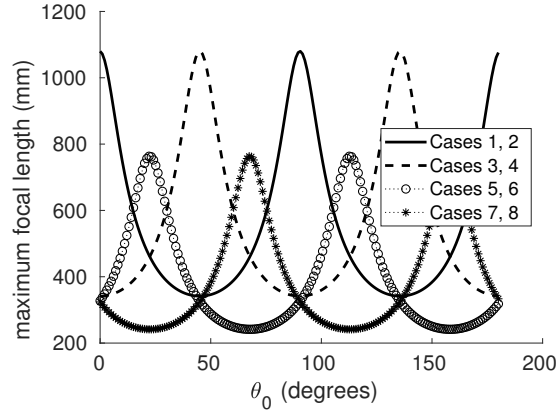


Figure 12: Simulation results for ORBIT motion type: f_{max} against θ_0 .

Finally, cases 7 and 8 produce similar results, since the target again moves diagonally to the TCS axes. However, when compared to cases 5 and 6, the perpendicularity of the motion directions leads to a phase difference of $\frac{\pi}{4}$.

4. Shot Type Feasibility

In cinematography planning, it is important to be able to determine whether a desired shot type is feasible, given a specific camera motion type and the target's physical dimensions. The shot type is primarily defined by the ratio of the target ROI height to the video frame height, therefore, it is linked to the video frame area being covered by the target ROI. Thus, below, video frame coverage refers to the ROI-to-video-frame-height ratio.

In order to examine the feasibility of a shot type, the appropriate focal length f_s leading to the desired target video frame coverage must be calculated. For motion types where the distance between the target and the UAV varies over time, keeping a constant target video frame coverage by constantly adjusting the camera focal length simulates the cinematographic “dolly zoom” effect [5].

The shot type can be achieved without risking 2D visual tracking failure, if the following relation holds:

$$f_s \leq f_{max} \quad (46)$$

In order to calculate the appropriate f_s for achieving the shot types described in Section 2 with respect to the desired UAV/camera motion type, we model the target as a sphere, with its center located at the TCS point $[0, 0, 0]^T$ and having constant radius R_t . Simple sphere-modelling allows us to consider its image on the video frame as a circle, with no perspective distortion when $\mathbf{l}_t = [0, 0, 0]^T$.

This rather simplistic target volume modelling facilitates us in deriving closed forms for f_s , without much deviation from reality when the object is not very flattened. In the case of significantly flattened targets, which could be better modelled

with a rectangular parallelepiped, sphere-based modelling results in an overestimation of f_s . Then, a simple solution is to perform the same analysis considering three different sphere radii, i.e., one for each parallelepiped dimension, and use either their mean, their maximum or their minimum. However, in the case of human heads, which is very important in cinematic media imaging, simple bounding sphere-based modelling is already quite accurate.

Below, the deviation vector \mathbf{q}_t is assumed to be equal to $[0, 0, 0]^T$ for the desired f_s calculations. Thus, no target motion deviations are taken into consideration, since they do not significantly affect the resulting video frame coverage percentage.

4.1. Constant target video frame coverage

Determining the video frame coverage for every UAV/camera motion type would normally include projecting the target sphere onto the video frame, finding the corresponding radius of the projected circle and computing the resulting coverage. This requires a search for the radius of the projected circle. The parameters determining the video frame coverage are the distance between UAV/camera and target, the camera focal length f and the physical target dimensions. Thus, without loss of generality, instead of directly projecting the target onto the current image plane, we determine the video frame coverage as if the UAV/camera was positioned exactly above the target in an altitude equal to the actual distance between them. Thus, it is trivial to find a 3D point being projected on the target image circle. Then, the latter's radius is the distance between the projection of the above 3D point and the principal point. This projection can be obtained by Eqs. (22) and (23) in pixel coordinates. The corresponding continuous coordinates of x_{im} and y_{im} on the image sensor are given by:

$$x_{im} = x_d s_x, \quad y_{im} = y_d s_y. \quad (47)$$

Thus, the video frame coverage percentage for the circular target ROI is given by:

$$c_s = \frac{2R_{im}}{H s_y}, \quad R_{im} = \sqrt{x_{im}^2 + y_{im}^2}. \quad (48)$$

where H is the height of the video frame in pixels and s_y the physical height of one pixel.

The above equations can be further simplified by defining R_{im} as the perspective projection of $\mathbf{p}_r = [R_t, 0, 0]^T$ (in TCS), where R_t is target radius, and by positioning the UAV/camera at $\mathbf{x}' = \mathbf{x}_{t+1} = [0, 0, z_d]^T$ where $z_d = \sqrt{x_{t'1}^2 + x_{t'2}^2 + x_{t'3}^2}$ is the distance between the target and the camera. Then, $y_{im} = 0$, thus, $R_{im} = x_{im}$ and:

$$x_{im} = \frac{1}{2} c_s H s_y \quad (49)$$

By utilizing Eqs. (22) and (47), and setting $o_x = 0$:

$$x_{im} = -f_s \frac{\mathbf{r}_1(\mathbf{p}_r - \mathbf{x}')}{\mathbf{r}_3(\mathbf{p}_r - \mathbf{x}')}. \quad (50)$$

The rotation matrix in this case is described by Eq. (35), and the appropriate focal length can be obtained by:

$$f_s = \frac{c_s H s_y z_d}{2R_t}. \quad (51)$$

Table 3: Shot type feasibility for UAV/camera motion types with constant distance from the target.

Motion type	$\min f_{max}$	f_s , when $c_s = 25\%$	f_s , when $c_s = 85\%$
LTS	194.4 mm	78.57 mm	267.14 mm
CHASE	142.4 mm	78.57 mm	267.14 mm
ORBIT	241.5 mm	78.57 mm	267.14 mm

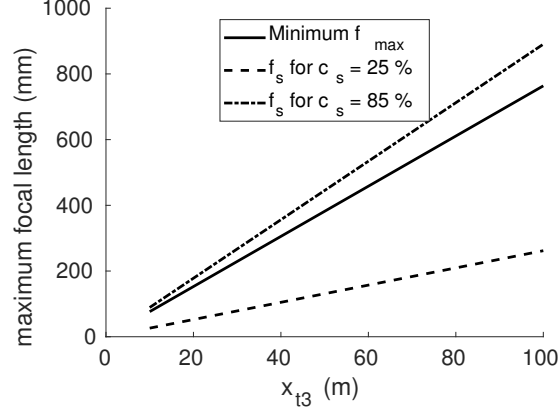


Figure 13: Maximum focal length f_{max} and f_s for Medium Shot and Close-Up Shot, against various UAV altitude, for VTS.

4.2. Simulations for constant target video frame coverage

In order to investigate the target tracking feasibility for specific shot type-UAV/camera motion type combinations, one can repeat the simulations described in Section 3.2 and determine if the desired f_s is below the minimum value of f_{max} for all cases. A trivial addition, which is omitted here for brevity, would include a check for violations of lens-specific upper/lower focal length limits.

For the UAV/camera motion types where the distance between the camera and the target remains constant (i.e., CHASE, ORBIT, LTS), the desired f_s is also constant for the entire shot. On the contrary, when the distance between the target and the UAV/camera varies (i.e., MAPMT, MATMT, FLYBY, FLYOVER, VTS), the appropriate f_s varies correspondingly. Although VTS is normally a UAV/camera motion type where the distance between the UAV and the target remains constant, it was studied for varying z_d in our simulations. Hence, in the first group of camera motion types, shot feasibility can be determined simply by two values, the minimum f_{max} and the desired f_s . In the second group, feasibility should be examined for the entire shot duration, or for a range of z_d values in the case of VTS.

For simulation purposes, we assume a sphere-shaped target positioned in $\mathbf{p} = [0, 0, 0]^T$ (in TCS), with radius $R_t = 1$ m (e.g., a racing bicycle during sports event coverage). In all motion types, the UAV and target position/motion/deviation properties comply with the descriptions in Section 3.2. In addition, the video frame resolution was set to $W = 1280$ pixels and $H = 720$ pixels. Simulations were carried out for two desired video frame coverage percentages, i.e., $c_s = 25\%$ and $c_s = 85\%$, corre-

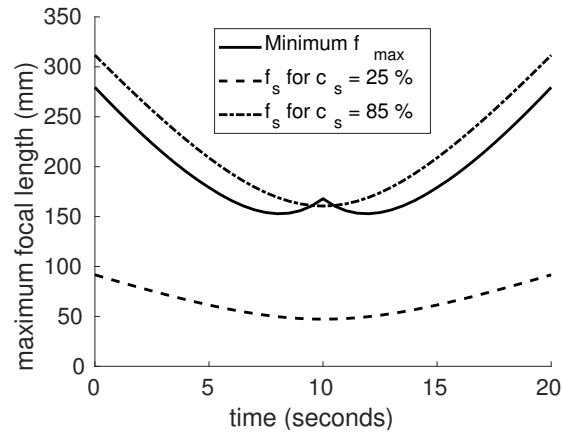


Figure 14: Maximum focal length f_{max} and f_s for Medium Shot and Close-Up Shot, against time t , for FLYBY.

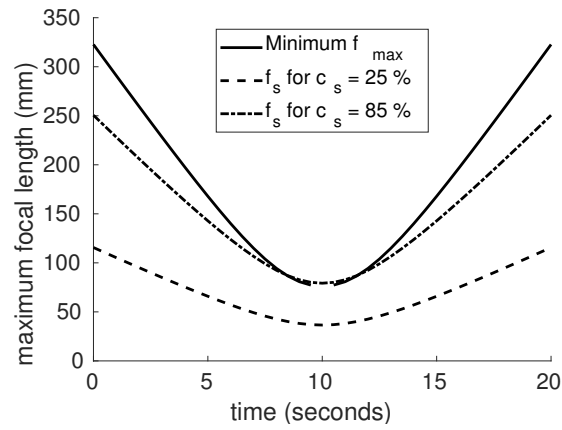


Figure 15: Maximum focal length f_{max} and f_s for Medium Shot and Close-Up Shot, against time t , for FLYOVER.

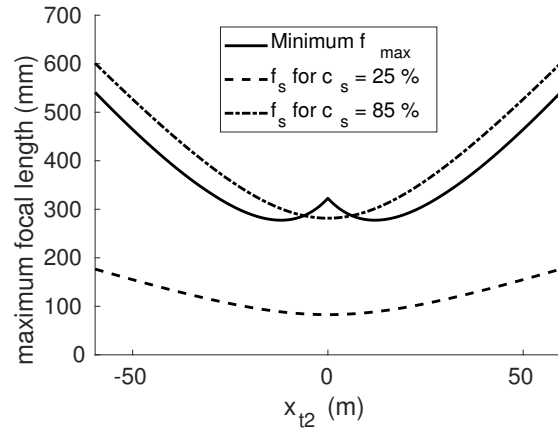


Figure 16: Maximum focal length f_{max} and f_s for Medium Shot and Close-Up Shot, against various UAV positions, for MAPMT.

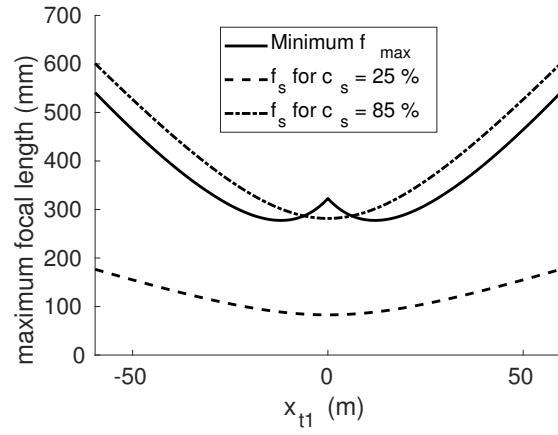


Figure 17: Maximum focal length f_{max} and f_s for Medium Shot and Close-Up Shot, against various UAV positions, for MATMT.

sponding to a Long Shot and a Close-Up Shot, respectively. Table 3 indicates that a Long Shot is achievable for the UAV/camera motion types CHASE, ORBIT and for LTS, while a Close-Up not feasible for any of these motion types.

For VTS, FLYBY, FLYOVER, MAPMT and MATMT the results are presented in Figures 13, 14, 15, 16 and 17 respectively. In these motion types, a Long Shot is achievable at all times ($f_s < f_{max}$), but a Close-Up could cause visual tracking failure in the presence of target velocity deviations.

The simulation results lead to the conclusion that 2D visual tracking of a real target is indeed a fairly challenging task at greater zoom levels, if the target deviates non-negligibly from the expected position on the next video frame.

4.3. Maximum permissible velocity deviation vector

By inverting the analysis made for f_{max} and fixing focal length to the f_s needed for a specific shot type, we can define the maximum permissible norm of the target velocity deviation vector $\mathbf{q}_t = [q_{t1}, q_{t2}, 0]^T$. This way, one can pre-determine whether a shot type is feasible from known/expected target/target route characteristics.

Below, we assume for simplicity that:

$$q_t = q_{t1} = q_{t2}, \quad (52)$$

to demonstrate the process. By denoting $t' = t + 1$, then q_t is given by solving the following equation, derived from Eq. (34):

$$(f_s^2 D_q - A_q^2 B_q^2) q_t^2 + 2A_q^2 B_q C_q q_t - A_q^2 C_q^2 = 0, \quad (53)$$

where $A_q = R_{max} d_{t'} s_x s_y$, $B_q = x_{t'1} + x_{t'2}$, $C_q = F \|\mathbf{x}_{t'}\|^2$ and $D_q = s_x^2 x_{t'3}^2 B_q^2 + s_y^2 \|\mathbf{x}_{t'}\|^2 (x_{t'1} - x_{t'2})^2$.

When $q_t > 0$, as in case 5 of the performed simulations, q_t can be directly obtained by:

$$q_t = \frac{A_q F \|\mathbf{x}_{t'}\|^2}{f_s \sqrt{D_q} + A_q (x_{t'1} + x_{t'2})}. \quad (54)$$

The maximum q_t can be obtained similarly for other cases and UAV/camera motion types, in order to estimate the range of permissible target velocity deviations for a specific shot type-UAV/camera motion type combination.

4.4. AirSim simulations for evaluating shot feasibility rules

In order to evaluate the presented shot feasibility rules under actual media production conditions, a realistic simulation was developed that implements the platform setup discussed thus far and incorporates the proposed rules. To this end, AirSim [33] was employed, i.e., an open source, highly realistic UAV simulation environment (based on the Unreal 4 real-time 3D graphics engine). For the evaluation purposes two different scenarios were developed (bike and track and field scenarios). In both scenarios, the generated shots involve a moving target (cyclist or running athlete) and a UAV equipped with a cinematographic camera, controlled by an API script, that follows the target according to the desired shot type/camera motion type combination. Snapshots



Figure 18: Snapshot from the synthetic, realistic evaluation environment. The UAV follows the target (bicycle) while performing an ORBIT motion type. The focal length of the camera is set to $50mm$, resulting in a Long Shot shot type.



Figure 19: Snapshot from the scenario in the synthetic, realistic evaluation environment. The UAV follows a running athlete while performing an ORBIT motion type.

from the generated footage are depicted in Figures 18 and 19, while an example 2D plot of the target and UAV trajectories, during an ORBIT, are shown in Figure 20.

The various parameters (e.g., focal length, UAV height, initial position relative to target etc.) were set similarly to the evaluation in Section 3.2. R_{max} was set adaptively to $\min(\frac{1}{2}H, \frac{wk}{s_y}R_{im})$, where the latter term is the search region size, defined by the 2D target ROI radius (in pixels) $\frac{1}{s_y}R_{im}$, a constant scaling factor w (set here to 1.5, as is the default value in [12]) and a varying scaling factor $k \in [0, 1]$ that shrinks the search region according to the proximity of the current ROI to the video frame borders, so as to restrict out-of-frame ROI translations that would cause 2D tracker drift and gimbal control failure.

Datasets created in such a manner can produce fully accurate results for both the target and UAV 3D location. However, this is not in line with a real-world scenario involving noisy GPS sensors. Thus, the 3D positions of both the target and UAV for every time instance t were distorted according to a Gaussian noise distribution, so as to simulate GPS measurements.

The experiments were carried out for all motion types, while attempting to achieve three different shot types: Long Shot (LS), Medium Close-Up (MCU) and Close-Up

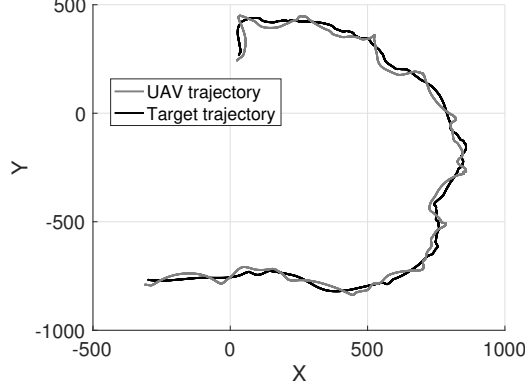


Figure 20: 2D plot of the UAV and target trajectories in WCS, during an ORBIT session in the AirSim simulator.

(CU). For evaluation purposes, we obtained the noisy 3D positions of both the target and the UAV at every time instance t . Additionally, the previous noisy 3D position of the target (from time instance $t - 1$) was employed to calculate its velocity. Assuming that the target will follow momentarily a linear trajectory, we estimate its 3D position in the next time instance ($t' = t + 1$) and adjust the UAV motion, so that the desired central composition framing is maintained. Then, at time instance t' , we compare the 2D projection of the estimated 3D target position with the 2D projection of the ground-truth 3D target position. If the distance of the two ROI center points, R_f , is above the R_{max} limit, ground-truth tracking failure is assumed ($R_f > R_{max}$). This is then compared with the predictions of Eqs. (32) for the current maximal permissible focal length and (51) for the desired one, regarding the current shot's feasibility, given the noisy 3D positions of the target and the UAV, the calculated target velocity and the estimated target position on the next video frame. By employing the above the proposed method assumes tracking failure when the desired focal length given by Eq. (51) is greater than the result of Eq. (32), as described by Eq. (46). The velocity deviation vector \mathbf{q}_t in Eq. (32) is simply calculated as the difference between the estimated target velocity at time instance $t - 1$ and the actual target velocity at time instance t (distorted by noise). Therefore, a reasonable assumption of temporally localized constant target acceleration is made. Thus, true/false positive/negative prediction labels (TP, FP, TN, FN) are computed for each time instance. Then, precision is calculated as $P = \frac{TP}{TP+FP}$, recall rate $R = \frac{TP}{TP+FN}$ and F-Measure as $F = \frac{2TP}{2TP+FP+FN}$.

In the first evaluation scenario of cycling, the mean precision, recall and F-Measure of the proposed rules over all motion types were 0.929, 0.994 and 0.960, respectively. Table 4 depicts the evaluation results per shot type, while Figure 21 contains the F-Measure box-plots for all motion types, separately for each shot type. In the second scenario of the running athlete, the mean precision, recall and F-Measure were 0.961, 0.927, 0.995 while the individual results per shot types are depicted in Table 5. Figure 22 demonstrates the F-Measure box-plots for all motion types in the second scenario,

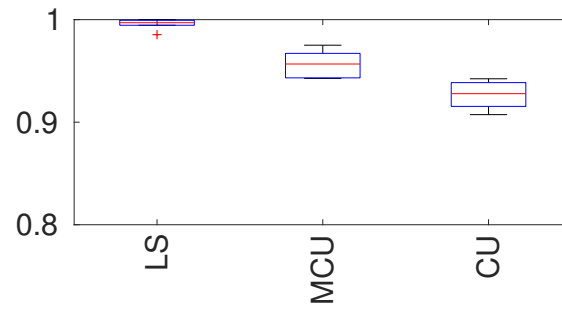


Figure 21: Box-plot of F-Measure for the three different shot types in the AirSim cycling evaluation test. The line inside the boxes demonstrates the median value in each case. Overall, CHASE performed the best and FLYOVER the worst.

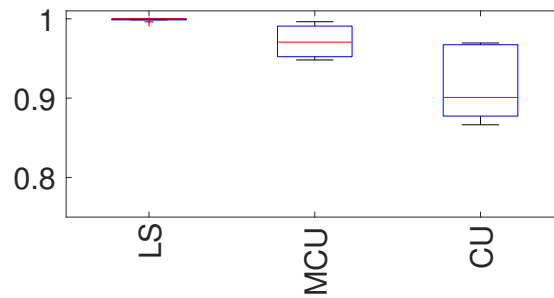


Figure 22: Box-plot of F-Measure for the three different shot types in the AirSim track and field evaluation test. The line inside the boxes demonstrates the median value in each case. Overall, VTS performed the best and LTS the worst.

Table 4: Mean evaluation results for the proposed shot feasibility rules over all motion types, in the realistic AirSim cycling setup.

Shot type	F-Measure	Precision	Recall
LS	0.992	0.991	0.997
MCU	0.956	0.923	0.993
CU	0.926	0.872	0.990
Mean	0.960	0.929	0.994

Table 5: Mean evaluation results for the proposed shot feasibility rules over all motion types, in the realistic AirSim track and field setup.

Shot type	F-Measure	Precision	Recall
LS	0.999	0.991	0.997
MCU	0.971	0.944	0.991
CU	0.913	0.845	0.999
Mean	0.961	0.927	0.995

separated per shot type.

In addition, the target ROI size calculation methodology was evaluated. As already mentioned, we treat the target as a sphere-shaped object in order to derive the desired focal length f_s . This can lead to approximation errors in video frame coverage estimation, especially with flattened targets. The focal length necessary to keep the desired shot type was calculated for each video frame, using the noisy 3D UAV and target positions, as well as the target ROI prediction for the next video frame.

The actual ROI-to-video-frame-height ratio was calculated at each time instance and compared with the desired value of c_s , as defined by each shot type. Figure 23 depicts the distribution of the actual video frame coverage vs the estimated one. Despite variations in the actual target ROI size, the proposed f_s calculation manages to keep the estimated target ROI size within the video frame coverage range of the desired shot type. Table 6 demonstrates the mean video frame coverage values for the three eval-

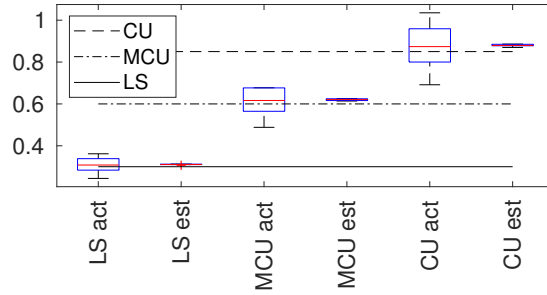


Figure 23: Box-plot of the estimated vs the actual target video frame coverage for the three desired framing shot types. Despite the simple sphere-based target modeling and the target/UAV localization noise, the estimated target ROI size lies within the range of the same shot type as the actual target ROI size.

Table 6: Desired, actual and estimated mean video frame coverage.

Shot type	Desired c_s	Actual c_s	Estimated c_s
LS	0.3	0.307	0.310
MCU	0.6	0.606	0.620
CU	0.85	0.872	0.880

uated shot types, over all the simulated motion types. Desired c_s is the video frame coverage percentage requested by the director, actual c_s is the video frame coverage percentage achieved by the produced ROIs, while estimated c_s refers to the coverage percentage that would be achieved if ground-truth, non-noisy UAV and target 3D positions were available. The largest deviation is observed in the CU case where, as already demonstrated in Section 4, target tracking is not feasible most of the time.

5. Conclusions

In this paper, a close examination of the shot type constraints arising in computer vision-assisted UAV active target following for cinematography applications has been performed. To this end, a number of industry-standard target-tracking UAV motion types have been strictly defined and geometrically modelled, while compatible shot types have been identified for each case. Subsequently, maximum permissible camera focal length, so that 2D visual tracking does not fail, as well shot type feasibility conditions were analytically determined. The relevant derived formulas can be readily employed as low-level rules in UAV intelligent shooting and cinematography planning systems. Practical simulations showcase the validity of our findings, since results comply with intuitive expectations in all cases.

Several extensions can be envisioned for the proposed rules. For instance, tighter integration with a specific real-time 2D visual tracker may lead to improvements. Additionally, since our formulas rely on the estimated velocity deviation vector \mathbf{q} at each time instance, learning to predict this vector from visual data (e.g., expected target route) would be a promising avenue for future research. Such a prediction may concurrently benefit the 2D visual tracker itself, as in [17] [39].

6. Acknowledgement

Funding: The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors’ views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

- [1] Computational UAV cinematography for intelligent shooting based on semantic visual analysis.

- [2] J. Angeles. *Fundamentals of robotic mechanical systems*, volume 2. Springer, 2002.
- [3] I. Arev, H. S. Park, Y. Sheikh, J. K. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics*, 33(4): 81, 2014.
- [4] S. Bhattacharya, R. Mehran, R. Sukthankar, and M. Shah. Classification of cinematographic shots using lie algebra and its application to complex event recognition. *IEEE Transactions on Multimedia*, 16(3):686–696, 2014.
- [5] B. Brown. *Cinematography: Theory and Practice: Image Making for Cinematographers and Directors*. Focal Press, 3rd edition, 2016.
- [6] P. Carr, M. Mistry, and I. Matthews. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 2013.
- [7] E. Cheng. *Aerial Photography and Videography Using Drones*. Peachpit Press, 2016.
- [8] L.-Y. Duan, J. S. Jin, Q. Tian, and C.-S. Xu. Nonparametric motion characterization for robust classification of camera motion patterns. *IEEE Transactions on Multimedia*, 8(2):323–340, 2006.
- [9] H. Fourati and D.E.C. Belkhiat. *Multisensor Attitude Estimation: Fundamental Concepts and Applications*. CRC Press LLC, 2016.
- [10] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Positioning Systems, inertial navigation, and integration*. John Wiley & Sons, 2007.
- [11] M. A. Hasan, M. Xu, X. He, and C. Xu. CAMHID: Camera motion histogram descriptor and its application to cinematographic shot classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1682–1695, 2014.
- [12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.
- [13] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [14] X. Huang, R. Janaswamy, and A. Ganz. Scout: Outdoor localization using Active RFID technology. In *Proceedings of the IEEE Conference on Broadband Communications, Networks and Systems (BROADNETS)*, pages 1–10, 2006.
- [15] N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan. An interactive tool for designing quadrotor camera shots. *ACM Transactions on Graphics*, 34(6):238, 2015.

- [16] N. Joubert, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, and P. Hanrahan. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691*, 2016.
- [17] T. Li. Single-road-constrained positioning based on deterministic trajectory geometry. *IEEE Communications Letters*, 23(1):80–83, 2018.
- [18] N. Liang, G. Wu, W. Kang, Z. Wang, and D. D. Feng. Real-time long-term tracking with prediction-detection-correction. *IEEE Transactions on Multimedia*, PP(99):1–1, 2018.
- [19] C. Liu, P. Liu, W. Zhao, and X. Tang. Robust tracking and re-detection: Collaboratively modeling the target and its context. *IEEE Transactions on Multimedia*, 2017.
- [20] I. Mademlis, V. Mygdalis, C. Raptopoulou, N. Nikolaidis, N. Heise, T. Koch, J. Grunfeld, T. Wagner, A. Messina, F. Negro, S. Metta, and I. Pitas. Overview of drone cinematography for sports filming. In *European Conference on Visual Media Production (CVMP) (short)*, 2017.
- [21] I. Mademlis, V. Mygdalis, N. Nikolaidis, and I. Pitas. Challenges in Autonomous UAV Cinematography: An Overview. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2018.
- [22] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas. High-level multiple-UAV cinematography tools for covering outdoor events. *IEEE Transactions on Broadcasting*, 2019.
- [23] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina. Autonomous UAV cinematography: A tutorial and a formalized shot type taxonomy. *ACM Computing Surveys*, 2019. accepted for publication.
- [24] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina. Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments. *IEEE Signal Processing Magazine*, 36(1):147–153, 2019.
- [25] S. Minaeian, J. Liu, and Y.-J. Son. Effective and efficient detection of moving targets from a UAV’s camera. *IEEE Transactions on Intelligent Transportation Systems*, 2018.
- [26] P. P. Mohanta, S. K. Saha, and B. Chanda. A model-based shot boundary detection technique using frame transition parameters. *IEEE Transactions on Multimedia*, 14(1):223–233, 2012.
- [27] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for UAV tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016.
- [28] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *arXiv preprint arXiv:1610.06475*, 2016.

- [29] R. Mur-Artal and J. D. Tardós. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.
- [30] T. Nägele, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics*, 36(4):132:1–132:10, 2017.
- [31] P. Nousi, E. Patsiouras, A. Tefas, and I. Pitas. Convolutional neural networks for visual information analysis with limited computing resources. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.
- [32] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas. Embedded UAV Real-time Visual Object Detection and Tracking. In *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.
- [33] S. Shah, D. Dey, C. Lovett, and A. Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Proceedings of the Field and Service Robotics Conference*, 2017.
- [34] C. Smith. *The Photographer’s Guide to Drones*. Rocky Nook, 2016.
- [35] A. Torres-González, J. Capitán, R. Cunha, A. Ollero, and I. Mademlis. A multi-drone approach for autonomous cinematography planning. In *Proceedings of the Iberian Robotics Conference (ROBOT’)*, 2017.
- [36] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [37] I. Tsingalis, A. Tefas, N. Nikolaidis, and I. Pitas. Shot type characterization in 2D and 3D video content. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2014.
- [38] X. Wang, H. Zhu, D. Zhang, D. Zhou, and X. Wang. Vision-based detection and tracking of a mobile ground target using a fixed-wing UAV. *International Journal of Advanced Robotic Systems*, 11, 2014.
- [39] L. Xu, Y. Liang, Z. Duan, and G. Zhou. Route-based dynamics modeling and tracking with application to air traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [40] O. Zachariadis, V. Mygdalis, I. Mademlis, N. Nikolaidis, and I. Pitas. 2D visual tracking for sports UAV cinematography applications. In *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017.