Force from Motion: Decoding Control Force of Activity in a First-person Video

Hyun Soo Park and Jianbo Shi

Abstract—A first-person video delivers what the camera wearer (actor) experiences through physical interactions with surroundings. In this paper, we focus on a problem of *Force from Motion*—estimating the active force and torque exerted by the actor to drive her/his activity—from a first-person video. We use two physical cues inherited in the first-person video. (1) Ego-motion: the camera motion is generated by a resultant of force interactions, which allows us to understand the effect of the active force using Newtonian mechanics. (2) Visual semantics: the first-person visual scene is deployed to afford the actor's activity, which is indicative of the physical context of the activity. We estimate the active force and torque using a dynamical system that can describe the transition (dynamics) of the actor's physical state (position, orientation, and linear/angular momentum) where the latent physical state is indirectly observed by the first-person video. We approximate the physical state with the 3D camera trajectory that is reconstructed up to scale and orientation. The absolute scale factor and gravitation field are learned from the ego-motion and visual semantics of the first-person video. Inspired by an optimal control theory, we solve the dynamical system by minimizing reprojection error. Our method shows quantitatively equivalent reconstruction comparing to IMU measurements in terms of gravity and scale recovery and outperforms the methods based on 2D optical flow for an active action recognition task. We apply our method to first-person videos of mountain biking, urban bike racing, skiing, speedflying with parachute, and wingsuit flying where inertial measurements are not accessible.

Index Terms—First-person Vision, Physical Sensation, Optimal Control.

1 Introduction

Understanding human activities encompasses not only knowing 'what' we are doing, such as jumping, running, and cooking, but also 'how', i.e. recovering the underlying *controls* of actions through muscle movements. Most computer vision systems have been built on visual measurements from a camera looking at us from a third-person perspective such as surveillance cameras. This camera produces a view that often has a limited visual access to the muscle movements due to self-occlusion or low spatial resolution. Further, for many activities, such as wingsuit flying in Figure 1(a), recording the muscle movements requires active camera motion by following the actor, which is challenging in practice.

We tackle human activity understanding from a different perspective: a first-person video mounted on an actor's head or body. Our conjecture is that the physical state and control of the actor are encoded in her/his first-person video, which can be estimated by leveraging its visual and motion semantics. Yet, it is fundamentally challenging due to the inherent properties of first-person videos. a) Visibility: ironically, a first-person video can barely see the actor's body due to its limited field of view, i.e., the body kinematics cannot be measured; b) State observability: the ego-motion of the first-person video is produced by a resultant of multiple force/torque interactions where there exists an ambiguity of decomposing them into active forces controlled by the actor; (c) Geometric ambiguity: scenes are geometrically measured and reconstructed up to scale and orientation where Newtonian dynamics cannot be applied.

We address these challenges by studying *Force from Motion*—an algorithm that computes the active components of physical force and torque exerted by the actor. Specifically, our algorithm takes an input, a first-person sports video, and outputs the active force and torque in a physical metric space (e.g., force in N) aligned with the gravitational field. For instance, in a wingsuit flying first-person video[1], we compute not only where he traveled but also how he controlled by applying force and torque, e.g., momentum change along the roll axis to shift the heading direction as shown in Figure 1(b) and 1(c). To recover the actor's active components, we focus on decoding three dominant physical quantities from a first-person video: gravity, momentum, and force/torque.

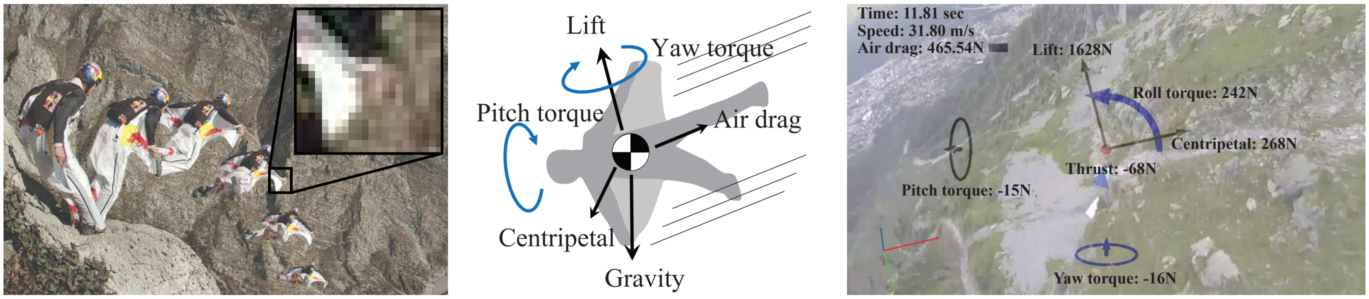
First, motion is strongly driven by the gravitational field which can be estimated from two visual cues in a first-person video. (1) Natural images encode the gravity direction because it affects how physical environment is formed, i.e. trees and buildings are usually vertical, water surface normal aligns with the gravity direction, and horizon is perpendicular to it. We learn such visual semantics of the gravity direction embedded in the first-person images. (2) The camera ego-motion is influenced by the gravity as the actor's activity always is accelerated along the gravity direction. We learn this relationship between camera trajectory and the gravity direction from training data.

Second, how fast we are going (speed or momentum) in the physical metric (m/s or kg·m/s) allows us to relate with how much force is applied through Newtonian dynamics. The absolute scale of our motion is revealed to us when the body is in a moment equilibrium during a banked turn, i.e., the balance between the gravity and centripetal force with respect to the leaning angle. We exploit the moment equilibrium to precisely compute the physical scale given the known gravitational constant, i.e., g =9.81 m/s^2.

Third, we decompose the force and torque into semantically

- H. S. Park is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 55455. E-mail: hspark@umn.edu
- J. Shi is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 19104. E-mail: jshi@seas.upenn.edu

1. https://www.youtube.com/watch?v=IM1vss7FXs8



(a) Third-person video (b) Physical model (c) First-person video

Fig. 1. (a) Extracting the control of actions from a third-person video is challenging due to limited visual accessibility to the muscle movements (occlusion and low resolution). Instead, this paper presents *Force from Motion* from a first-person video—inferring active components of physical force and torque to control the movement of the camera wearer (actor). (b) We recover the actor's physical state and control using a rigid body dynamics with an optimal control theory. (c) Our system produces the active force and torque in a gravitational field. As a by-product, the passive force such as air drag and lifting force can be recovered.

meaningful components, i.e., active components exerted by the actor (e.g., twisting body orientation in flying and pedaling in biking) and passive components exerted by the environment (e.g., air drag). We describe the actor's physical state (position, orientation, and linear/angular momentum) using a dynamical system where the latent state can be indirectly observed by the first-person video. We solve the dynamical system to compute the active components inspired by an optimal control thoery. As a by-product, we recover the passive forces such as air drag, friction, and ground reaction/lifting force. Our modeling is generic: different activities such as mountain biking, skiing, and speed flying can be modeled with a few minor modifications (e.g., mass and friction coefficient).

Why Egocentric Video? As a form factor of a video camera facilitates seamless integration into body, hundreds of thousands of egocentric videos are captured and shared via online video repositories such as YouTube, Vimeo, and Facebook. For instance, currently more than 6,000 GoPro videos are posted in YouTube in a day. Many of these videos capture speed sport activities such as downhill mountain biking (1-10 m/s), glade skiing (5-12 m/s), skydiving (60-80 m/s) from first-person view. These videos excite visual motion stimuli that are strongly dominated by physical sensation. Decoding such physical sensation provides a new computational representation of such videos that can be not only applied to vision tasks such as activity recognition, video indexing, content generation for virtual reality [54] but also computational sport analytics [49], sensorimotor learning [66], and sport product design [12].

A 6 DOF inertial sensor for body motion (IMU), strain gage for muscle tension, and pitot tube for air flow speed can measure the physical quantities associated with body dynamics. In spite of high sensitivity and precision, such sensors are not often integrated into a video recording activities, e.g., none of first-person videos in online repositories provides extra sensory data. Our system can predict such physical quantities without extra sensors that will augment a new dimension for understanding activities from first-person videos.

Contributions We build on an earlier version [41] of this paper, and the core contributions of this paper include: (1) Force from motion: we integrate rigid body dynamics into 3D reconstruction pipeline to estimate active force and torque by exploiting optimal control (i.e., an iterative formulation of linear quadratic regulator for first-person videos); (2) Gravity direction estimation: we learn visual gravity cues to predict 3D gravity direction using a sequence of image; (3) physical scale recovery: we recover a scale factor from the roll torque equilibrium relationship. We quantitatively evaluate our method using a controlled experiment with inertial measurement units (IMU). Our method shows quantitatively equivalent reconstruction comparing to IMU measurements in terms of gravity and scale recovery and outperforms the methods based on 2D optical flow for an active action recognition task. We apply our method to first-person videos of diverse activities such as mountain biking, urban bike racing, skiing, speedflying with parachute, and wingsuit flying.

2 Related Work

Understanding an internal model of physical interactions from visual data is key area of studies in psychology [4], neuroscience, robotics, and computer vision, e.g., learning visual sensorimotor skills [66]. This paper particularly focuses on decoding physical sensation from a first-person video by leveraging Newton's laws of motion. Such work is mostly done in third person videos, and in this section, we review most relevant work: modeling human motion and internal physics from third person videos, and learning visual semantics from first-person videos.

2.1 Human Behavior Modeling in 3rd Person View

Johansson's experiment [26] has shown that human motion can be perceived and predicted by a sparse representation with short duration of visual observation. However, enabling such perception for a machine is still challenging without prior knowledge due to a large degree of freedom of an articulated body structure. This requires a compact representation to describe human body motion. Three main representations have been studied: data driven, geometry based, and physics based representations.

Statistical models have shown strong discriminative power for high dimensional data such as human body motion. Sidenbladh et al. [55] maximized a poster distribution of joint angle by combining a prior of a kinematic chain and its likelihood from pixel intensity. Such Bayesian framework was extended by Choo and Fleet [10] that introduced an efficient sampling to approximate a posterior distribution of human pose in 3D. Urtasun et al. [61], [62] learned a motion prior by exploiting a subspace analysis which can cluster and track various motions. Howe et al. [21] learned a kinematic prior to resolve projective ambiguity, i.e., two 3D solutions exist given a monocular 2D image measurement as

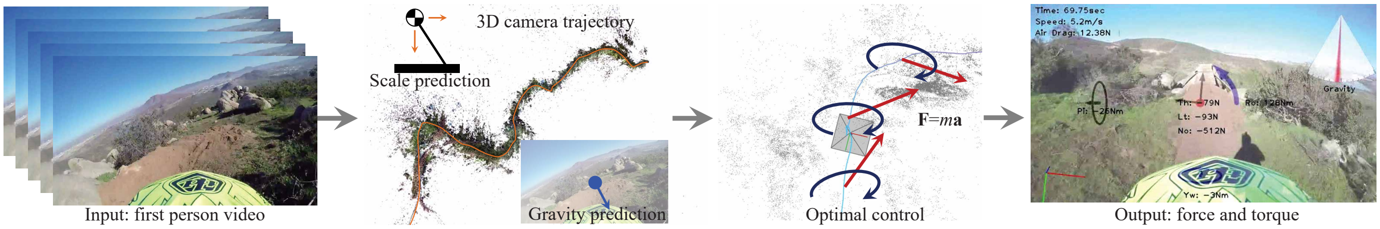


Fig. 2. Our system takes a first-person video of sporting activities and estimate the active force and torque that generates the camera ego-motion. We recognize a physical metric space by estimating gravity and metric scale. Based on the coordinate, we compute the optimal force acting on the actor's body by minimizing reprojection error.

noted by Taylor [58]. Other representations such as deformable part models [5] and a convolutional neural network [23] have been shown higher discriminative power that can be applied for real world scenes.

Bregler and Malik [8] modeled a kinematic constraint as a function of Lie group that allowed them to represent motion with a set of joint angles. Such parametrization is compact and therefore, suitable for action recognition task [47]. A factorization based approach [7], [60] was used by Yan and Pollefey [69] where they discovered a joint location and its type in an articulated structure using the fact that the joint space lies in an intersection between two subspaces spanned by two rigid bodies. Akhter and Black [2] exploited joint space limit conditioned by pose to reconstruct 3D human pose. The geometric approaches often combine with temporal constraints: Valmadre [63] used a temporal filter, and Akhter et al. [22] and Park et al. [43] used trajectory bases.

Metaxas and Terzopoulos [36] modeled motion and shape deformation from a video using Lagrangian mechanics. They integrated the equations of motion into a Kalman filtering framework to identify internal and external forces. A notable characteristics of their method is a capability to handle missing data due to occlusion, which is a critical issue in particular for a computer vision task. In a similar way, Wren and Pentland [67] proposed a direct control system utilizing Hidden Markov Models. Physics based approaches are often used for markerless motion capture: Brubaker et al [9] explicitly modeled the ground reaction force as an impulse function during bipedal walking. Wei and Chai [65] have shown a keyframe based human motion reconstruction where physics based simulation interpolates between keyframes. Vondrak et al. [64] introduced a feedback control system based on multi-body dynamics that provides a Bayesian prior to track human body motion.

2.2 First-person Vision

A first-person camera sees what the camera wearer sees, which differs from a third person system such as surveillance cameras, i.e., direct visual experiencing vs. observing at distance. This enables measuring subtle head movement, which has been a viable solution for behavior science and quality of life technology [27], [46], [48], and motivated many vision tasks such as understanding fixation point [33], identifying eye contact [70], and localizing joint attention [16], [42].

A first-person camera ego-motion is a highly discriminative feature for activity recognition. Fathi et al. [15], [16] used gaze and object segmentation cues to classify activities. 2D motion features were exploited by Kitani et al. [28] to categorize and segment a first-person sport video in a unsupervised manner. Coarse-to-fine motion models [52] and a pretrained convolutional neural network [53] provided a strong cue to recognize activities. Yonetani et al. [71] utilized a motion correlation between first and third person videos to recognize people's identity. Kopf et al. [29] stabilized first-person footage via 3D reconstruction of camera ego-motion. In a social setting, joint attention was estimated via triangulation of multiple camera optical rays [42], [44] and the estimated joint attention was used to edit social video footage [3].

Another information that the first-person camera captures is exomotion or scene motion. Pirsiavash and Ramanan [45] used an object centric representation and temporal correlation to recognize active/passive objects from a egocentric video, and Rogez et al. [50] leveraged a prior distribution of body and hand coordination to estimate poses from a chest mounted RGBD camera. Lee et al. [31] summarized a life-logging video by discovering important people and objects based on temporal correlation, and Xiong and Grauman [68] utilized a web image prior to select a set of good images from egocentric videos. Fathi et al. [16] used observed faces to identify social interactions and Pusiol et al. [46] learned a feature that indicates joint attention in child-caregiver interactions.

As the camera wearer interacts with surroundings, first-person videos can encode affordance of the scene. For instance, semantic meaning of scene 3D layout (e.g., building, road, and street signs) tells us about motion affordance, allowing predicting future activities [40], [56] and objects to interact [6]. Gaze direction can be precisely estimated from visual semantics and 3D motion of first-person videos [33], important objects can be detected [31], [45], visual transformation can be predicted through ego-motion [24], and robust feature can be learned [1].

Our approach: To our best knowledge, this is the first paper that provides a computational framework to understand a first-person video based on physical body dynamics. As an egocentric video has limited observation of body parts, estimating force and its control significantly differs from previous problems of physics based tracking and reconstruction.

3 Overview

Our algorithm takes an input, a first-person video of sporting activities, and outputs the active force and torque that generate the actor's first-person video as illustrated in Figure 2. We use a dynamical system (Section 4) to model the actor's physical state transition where its dynamics is described by an inverted pendulum model (Section 5). Her/his physical state is mapped to the first-person visual motion (Section 6) with the estimated physical scale and gravity direction. We solve the dynamical system using a linear quadratic regulator by minimizing reprojection error (Section 7).

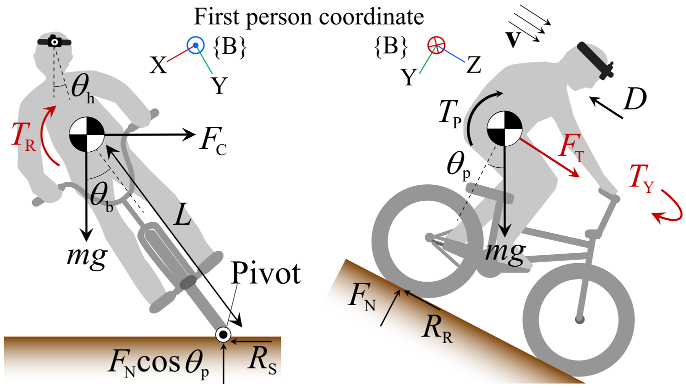


Fig. 3. We model the actor's dynamics using an inverted pendulum where the force and torque are decomposed into two components: passive components (weight, mg; centripetal force, $F_{\rm C}$; normal force, $F_{\rm N}$; sliding and rolling friction force, $R_{\rm S}$ and $R_{\rm R}$; air drag, D; pitch torque, $T_{\rm P}$) and active components (thrust, $F_{\rm T}$; roll torque, $T_{\rm R}$; yaw torque, $T_{\rm Y}$).

4 Dynamical System for First-person Video

We model the dynamics of the actor in a first-person video using a discrete nonlinear dynamical system [57]:

$$\mathbf{y}_{t+1} = f_{\rm dyn}(\mathbf{y}_t, \mathbf{u}_t) \tag{1}$$

$$\mathcal{I}_t = f_{\rm prj}(\mathbf{y}_t; \mathcal{S}), \tag{2}$$

where $f_{\rm dyn}$ maps a transition of the actor's latent physical state $\mathbf{y}_t$ (e.g., position, orientation, velocity) given the actor's control input $\mathbf{u}_t$ (i.e., active force and torque). $f_{\rm prj}$ generates the actor's first-person image $\mathcal{I}_t$ by projecting a 3D visual scene $\mathcal{S}$ to the camera, which links the actor's physical state to his/her visual scene. The goal of the paper is to estimate $\{\mathbf{u}_t\}$ given the sequence of images $\{\mathcal{I}_t\}$ from the first-person video.

Computing $\{\mathbf{u}_t\}$ directly from Equation (1) and (2) is intractable due to unknown $f_{\rm dyn}$, $f_{\rm prj}$, and $\mathcal{S}$. In the subsequent sections, we derive the dynamical system ($f_{\rm dyn}$ in Section 5 and $f_{\rm prj}$ in Section 6) and present an algorithm to solve it in Section 7.

5 Dynamics of Actor

We approximate the dynamics of the actor, $f_{\rm dyn}$ in Equation (1) using a 3D inverted pendulum model as shown in Figure 3. The force and torque are represented in the first-person body coordinate, $\{\mathcal{B}\}$. We define the actor's physical state and active force and torque (control input) as follows:

$$\mathbf{y} = \left[\begin{array}{cc|cc} \mathbf{C}^{\mathsf{T}} & P & \mathbf{q}^{\mathsf{T}} & \mathbf{L}^{\mathsf{T}} \end{array}\right]^{\mathsf{T}} \in \mathbb{R}^{11} \tag{3}$$

$$\mathbf{u} = \left[\begin{array}{ccc} F_{\rm T} & T_{\rm Y} & T_{\rm R} \end{array}\right]^{\mathsf{T}} \in \mathbb{R}^3, \tag{4}$$

where $\mathbf{C} \in \mathbb{R}^3$ and $\mathbf{q} \in \mathbb{S}^3$ are the 3D location and orientation (quaternion) of the actor's center of mass, P is the linear momentum along the instantaneous velocity, $P = m\|\mathbf{v}\|$, and $\mathbf{L}$ is the angular momentum, $\mathbf{L} = \mathbf{R}^{\mathsf{T}}\mathcal{J}^{-1}\mathbf{R}\boldsymbol{\omega}$. $\mathbf{v}$ is the linear velocity, $\mathcal{J}$ is moment of inertia, and $\boldsymbol{\omega}$ is angular velocity in the first-person coordinate. $\mathbf{R} \in SO(3)$ is a matrix representation of $\mathbf{q}$. The active force and torque (control input) $\mathbf{u}$ are composed of three elements: $F_{\rm T}$ is the thrust force applied along the velocity direction using pedaling and braking actions, $T_{\rm Y}$ is the yaw torque applied through steering wheel, and $T_{\rm R}$ is the roll torque to balance the posture.

Net force and torque act on the actor's center of mass, which affects the physical state through linear and angular acceleration. The actor's body is pivoted at the ground contact point (Figure 3), which forms an inverted pendulum model. The equation of motion can be written as:

$$f_{\rm dyn}(\mathbf{y}, \mathbf{u}) = \left[\begin{array}{c} \mathbf{C} + \mathbf{R}^3 P\Delta t/m \\ P + (F_{\rm R} + F_{\rm T})\Delta t \\ \mathbf{q} + (\mathcal{J}^{-1}\mathbf{L})\mathbf{q}/2 \\ \mathbf{L} + \left[\begin{array}{c} 0 \\ T_{\rm Y} \\ \tau_{\rm R} + T_{\rm R} \end{array}\right]\Delta t \end{array}\right], \tag{5}$$

where m is mass, and

$$F_{\rm R} = -R_{\rm R} - D + mg\mathbf{G}^{\mathsf{T}}\mathbf{R}^3$$
$$\tau_{\rm R} = -mgL\mathbf{G}^{\mathsf{T}}\mathbf{R}^1 - F_{\rm C}L\mathbf{G}^{\mathsf{T}}\mathbf{R}^2.$$

$\mathbf{G} \in \mathbb{S}^2$ is the 3D gravitational field direction, $g = 9.81$ m/s^2 is the gravitational constant. Along the Z direction, $R_{\rm R} = \mu_{\rm R} mg$ is the rolling/sliding friction along the Z axis of the first-person coordinate, and $\mu_{\rm R}$ is rolling friction coefficient. $D = 0.5C_{\rm D}\rho A\|\mathbf{v}\|^2$ is the air drag force where $C_D \approx 1.0$, $\rho = 1.23$ kg/m^3, and A are air drag coefficient, air density, and cross sectional area perpendicular to the velocity, respectively. Along the roll direction, $\tau_{\rm R}$ is the roll torque, L is the length from the pivot (i.e., ground contact point) to the actor's center of mass, and $F_{\rm C}$ is the centripetal force. $\mathbf{R}^i$ is the $i^{\rm th}$ row of $\mathbf{R}$.

Using the action-reaction relationship, we can compute the passive force and torque:

$$F_{\rm N} = mg\sin\theta_{\rm p} = mg\mathbf{G}^{\mathsf{T}}\mathbf{R}^2$$
$$R_{\rm S} = \mu_{\rm S} F_{\rm N}\cos\theta_{\rm p}$$
$$T_{\rm P} = -\mathbf{L}^1/\Delta t$$

where $\theta_{\rm p} = \cos^{-1}\left(\mathbf{G}^{\mathsf{T}}\mathbf{R}^3\right)$ is the pitch angle, and $\mathbf{L}^1$ is the angular momentum along the pitch direction. $F_{\rm N}$, $R_{\rm S}$, and $T_{\rm P}$ are normal or lifting force, sliding friction with $\mu_{\rm S}$ friction coefficient, and passive torque along the pitch direction created by an unbalance impact between two wheels in a bicycle as shown in Figure 3. Note that the biking activity is used for an illustrative purpose while this dynamics can generalize for various sporting activities such as skiing, jetskiing, speedflying, and wingsuit flying with a few minor modifications of coefficients such as body mass, moment of inertia, and air lift instead of normal force for a flying activities. See Appendix D for the activity dependent coefficients.

6 Visual Map to Actor's Physical State

Equation (2) describes the relationship between the actor's first-person video and physical state. We model the relationship by making two assumptions:

Assumption 1. The 3D trajectory of a first-person camera approximates that of the actor's center of mass.

This assumption allows linking the actor's physical state to the first-person video through its camera projection matrix, i.e., $\mathbf{P}(\mathbf{y}_t) = \mathbf{K}\mathbf{R}_t \left[\begin{array}{cc} \mathbf{I}_3 & -\mathbf{C}_t \end{array}\right] \in \mathbb{R}^{3\times4}$ where $\mathbf{K}$ is the intrinsic parameter of the first-person camera encoding focal length and principal points. We validate this assumption in Section 9.

Assumption 2. A set of 3D sparse feature points approximate the 3D geometry of the scene.

Fig. 4. We show the likelihood given an image with the red heatmap. The dotted lines are the ground truth gravity direction. The per pixel evidence [34] is encoded as transparency, i.e., the stronger evidence, the more transparent. The CNN correctly predicts gravity direction while the last image produces 15 degree error due to the tilted bicycler.

This assumption allows utilizing structure from motion [19] to model the actor's 3D visual scene, i.e., $\mathcal{S} \approx \{\mathbf{X}_p\}_{p=0}^{P}$ where $\mathbf{X} \in \mathbb{R}^3$ is a reconstructed 3D point.

With these assumptions, Equation (2) can be rewritten as:

$$\widetilde{\mathbf{x}}_{t,p} = f_{\text{prj}}(\mathbf{P}(\mathbf{y}_t), \mathbf{X}_p) \propto \mathbf{P}(\mathbf{y}_t)\widetilde{\mathbf{X}}_p, \tag{6}$$

where $\widetilde{\cdot}$ is a homogeneous representation of $\cdot$, and $\mathbf{x} \in \mathbb{R}^2$ is the 2D projected point of the 3D point.

Structure from motion enables decomposing Equation (6) into the 3D visual scene and the actor's physical state $\mathbf{y}$. However, this geometric decomposition involves a fundamental ambiguity, i.e., the 3D reconstruction is defined up to **scale** and **orientation**. There exists an arbitrary similarity transformation such that $\mathbf{P}\widetilde{\mathbf{X}} \propto c\mathbf{P}_a\mathbf{T}^{-1}\mathbf{T}\widetilde{\mathbf{X}}_a$ where $\mathbf{T} \in SE(3)$ is an Euclidean transform and c is scalar. This ambiguity precludes us from applying the actor's dynamics (Equation (5)) on the reconstructed $\mathbf{y}_a$ because the physical quantities are not represented in metric unit, e.g., distance in m, force in N, torque in Nm, and angle with respect to the gravitational field (Figure 4). In the following subsections, we leverage visual and motion semantics associated with first-person videos to estimate the orientation (gravitational field) and physical scale (metric units).

6.1 Gravitational Field Estimation

The gravitational field is a dominant physical quantity that drives motion, i.e., potential energy is converted to kinetic energy. There exist two gravity cues in first-person videos. (1) Visual semantics: a natural image encodes the gravity direction because it affects how physical environment is formed, i.e. trees and buildings are usually vertical, the horizon is perpendicular to gravity direction, and the water surface normal is aligned with it [18], [39]. It is possible to learn such visual semantics from first-person images to recognize the gravity direction. (2) Motion semantics: the actor's ego-motion is highly affected by the gravitational field, e.g., the body is often oriented upright, the forward ego-motion is driven by gravity, and the banked angle forms when centripetal force applied (turning). Thus, the 3D reconstructed camera trajectory can be used to recognize the gravity direction. We leverage these two semantic cues to estimate the 3D gravitation field from a sequence of first-person images.

We represent the gravitation field using a global unit vector in 3D, $\mathbf{G}(\phi_1, \phi_2) = \begin{bmatrix} \sin\phi_1\cos\phi_2 & \sin\phi_1\sin\phi_2 & \cos\phi_1 \end{bmatrix}^\mathsf{T} \in \mathbb{S}^2$ where a point in a unit sphere parametrized by polar ϕ_1 and azimuthal ϕ_2 angles in a spherical coordinate. Note that this gravity vector is represented in a global (world) coordinate, which applies to the entire images in the first-person video.

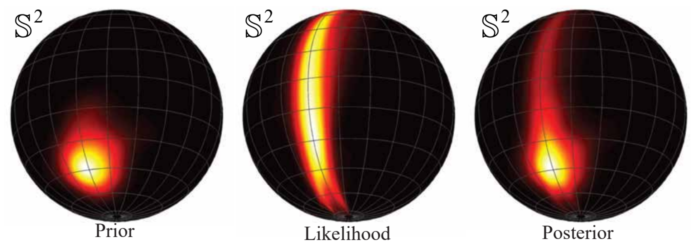


Fig. 5. We compute a maximum a posteriori estimate of the 3D gravity direction. We model the prior using a mixture of von Mises-Fisher distributions and learn a likelihood using a convolutional neural network.

We estimate the 3D gravity direction from a sequence of images using the maximum a posteriori (MAP) by fusing the visual and motion cues:

$$\begin{aligned}\mathbf{G}^* &= \underset{\mathbf{G}\in\mathbb{S}^2}{\operatorname{argmax}}\; p(\mathbf{G}|\mathcal{I}_1, \cdots, \mathcal{I}_T, \mathbf{R}_1, \cdots, \mathbf{R}_T) \\ &= \underset{\mathbf{G}\in\mathbb{S}^2}{\operatorname{argmax}}\; p_{\text{mot}}(\mathbf{G}|\mathbf{R}_1 \cdots \mathbf{R}_T)\prod_{t=1}^{T} p_{\text{vis}}(\mathcal{I}_t|\mathbf{G}, \mathbf{R}_t),\end{aligned} \tag{7}$$

where $p_{\text{mot}}(\mathbf{G}|\mathbf{R}_1 \cdots \mathbf{R}_T)$ is a gravity prior provided by the 3D reconstruction of the camera ego-motion, and $p(\mathcal{I}_t|\mathbf{G}, \mathbf{R}_t)$ is a likelihood computed by the visual semantics. $\{\mathbf{R}_t, \mathcal{I}_t\}$ is the sequence of images and their rotation matrices.

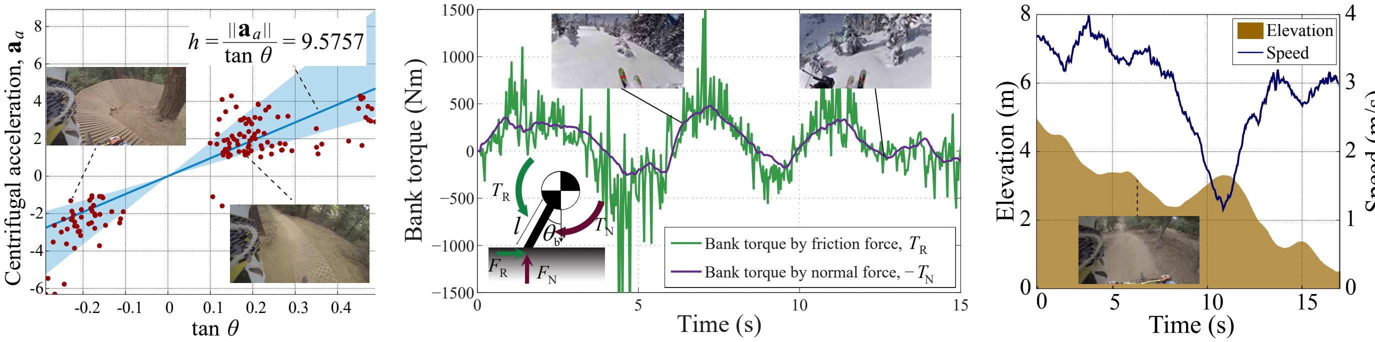
Figure 5 illustrates an MAP estimate of the gravitational field. The motion cue provides a prior distribution of the gravity where high probability forms near at the bottom of the unit sphere (left). Note that the visual semantic cue from a single image cannot predict the 3D gravity direction due to the information loss of 2D projection. Each image produces a streak in a likelihood distribution (middle)—any 3D gravity direction along the streak results in equivalent 2D gravity direction. This ambiguity can be further resolved by taking into account more images that moves different heading directions. The integration of multiple image predictions through Equation (7) collapses the streak into a unimodal distribution (right).

6.1.1 *Learning Gravity Likelihood from Visual Semantics*

We model the gravity likelihood from a first-person image by measuring how well the projected 3D gravity direction $\mathbf{G}$ agrees with its visual semantics:

$$p_{\text{vis}}(\mathcal{I}_t|\mathbf{G}, \mathbf{R}_t) = \mathcal{L}_{\text{vis}}\left(\mathcal{I}_t \,\middle|\, \theta_g = \tan^{-1}\frac{(\mathbf{R}_t^1)^\mathsf{T}\mathbf{G}}{(\mathbf{R}_t^2)^\mathsf{T}\mathbf{G}}; \mathbf{w}_{\text{vis}}\right).$$

where $\mathcal{L}_{\text{vis}}$ is a gravity probability distribution over the orientation θ_g, i.e., $\theta_g = 0$ if the gravity is aligned with the Y axis of the



(a) Scale factor (b) Dynamic balance (c) Elevation

Fig. 6. (a) We plot the centripetal acceleration computed by structure from motion with respect to the banked angle where its slope (blue line) $|\mathbf{a}_a|/\tan\theta_\mathrm{b}$ is the scale factor. (b) We verify the moment balance at a banked turn, $T_\mathrm{N} + T_\mathrm{R} = 0$. (c) The recovered gravity direction and scale allow us to compute the terrain elevation and speed in metric units.

image. $\mathbf{R}^1$ and $\mathbf{R}^2$ are the first and second rows of the rotation matrix, $\mathbf{R}$.

We learn the weights $\mathbf{w}_\mathrm{vis}$ of $\mathcal{L}_\mathrm{vis}$ via supervised learning using a convolutional neural network (CNN). We cast this learning gravity semantics as an image classification problem where the class corresponds to the image orientation, i.e., we discretize angle with 1 degree resolution across $\theta_g \in [-\pi/2, \pi/2]$. We use the probability computed by the softmax of the FC8 layer of AlexNet [30] to compute the likelihood distribution. The network weights are refined given the ImageNet [51] pre-trained model where a resized image (320×180) is used as an input of the network. The details of training, annotation process, and data can be found in Appendix A.

Figure 4 illustrates the likelihood of the gravity direction learned by CNN as shown in the red heatmap and dark red triangle shows prediction θ_g. We also encode the per pixel evidence of the gravity prediction using a fully convolutional neural network [34] using transparency, i.e., the stronger evidence, the more transparent. The CNN correctly predicts gravity direction while the last image produces 15 degree error due to the tilted orientation of the bicycler.

6.1.2 Learning Gravity Prior from Camera Motion

Therefore, a sequence of 3D body orientations provide a strong motion cue to recover the gravitational field. We leverage the local relative transform with respect to the t^th image, $\mathbf{R}_t$ to infer the gravity:

$$\widehat{\mathbf{G}}_t = g_\mathrm{mot}(\mathbf{q}_t^{t+1}, \cdots, \mathbf{q}_t^{t+\Delta t}; \mathbf{w}_\mathrm{mot}), \tag{8}$$

where $\mathbf{q}_t^{t+i}$ is the relative rotation from the t^th image to $(t+i)^\mathrm{th}$ image, $\mathbf{R}_{t+i}\mathbf{R}_t^\mathsf{T}$ in quaternion representation. g_mot encodes the dynamics of first-person rotation over time to predict the gravity direction, $\widehat{\mathbf{G}}_t$ parametrized by $\mathbf{w}_\mathrm{mot}$. We learn $\mathbf{w}_\mathrm{mot}$ using a long short-term memory (LSTM) [20] with loss: $L_\mathrm{mot}(\widehat{\mathbf{G}}, \mathbf{G}_\mathrm{gt}) = 1 - \widehat{\mathbf{G}}^\mathsf{T}\mathbf{G}_\mathrm{gt}$ where $\mathbf{G}_\mathrm{gt}$ is the ground truth gravity direction from training data in Appendix A.

Given a sequence of predictions $\{\widehat{\mathbf{G}}_t\}_{t=1}^{T-\Delta t}$, we model a prior distribution of the 3D gravity direction in Equation (7) using a mixture of von Mises-Fisher distributions:

$$p_\mathrm{mot}(\mathbf{G}|\mathbf{R}_1, \cdots, \mathbf{R}_T) \approx \sum_{m=1}^{M} \frac{\kappa_m}{4\pi \sinh \kappa_m} \exp\left(\kappa_m \mathbf{G}^\mathsf{T}\mathbf{g}_m\right) \tag{9}$$

where $\{\mathbf{g}_m, \kappa_m\}$ is a set of modes and concentration parameters, and M is the number of modes. We learn $\{\mathbf{g}_m, \kappa_m\}$ from $\{\widehat{\mathbf{G}}_t\}_{t=1}^{T-\Delta t}$ using an Expectation-Maximization algorithm [14].

6.2 Physical Scale Recovery

The physical state $\mathbf{y}_a$ decomposed by structure from motion has an arbitrary scale, i.e., there exists an unknown scale factor, α that upgrades the scale free linear acceleration $\mathbf{a}_a$ to the physical scale $\mathbf{a} = \alpha\mathbf{a}_a$ in m/s^2 unit. We exploit the gravitational constant, $g = 9.81$ m/s^2, that is revealed to us during a banked turn[2].

At the banked turn, a moment equilibrium occurs between the gravity and centripetal force with respect to the banked angle, θ_b as shown in Figure 3:

$$Lmg\sin\theta_\mathrm{b} - LF_\mathrm{C}\cos\theta_\mathrm{b} = 0, \tag{10}$$

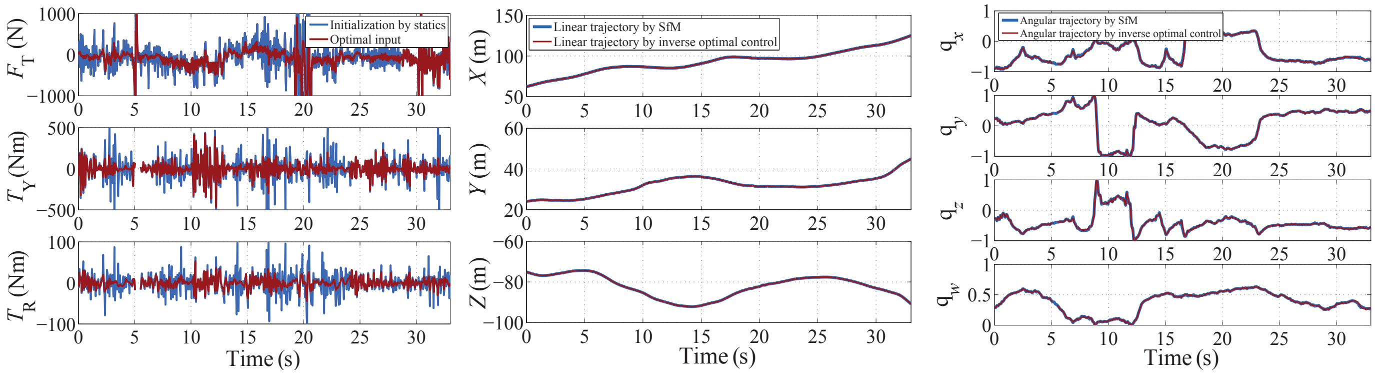
where θ_b is the banked angle. The lateral force is induced by centripetal acceleration, $F_\mathrm{C} = \alpha m\|\mathbf{a}_a\|$. Since g is constant, α can be solved as:

$$\alpha = \frac{g}{\|\mathbf{a}_a\|}\tan\theta_\mathrm{b}. \tag{11}$$

This moment equilibrium applies to not only the dynamics on the ground (biking and skiing) but also the aerodynamics (wingsuit flying) between the lifting force and centripetal force.

As shown in Figure 6(a), α can be computed when the moment equilibrium occurs (banked turn). The red points illustrate the centripetal acceleration, $\|\mathbf{a}_a\|$ with respect to the banked angle, $\tan\theta_\mathrm{b}$ where the point distribution form a line. The slope of the line is the scale factor, $1/\alpha$, and $\tan\theta_\mathrm{b} < 0$ and $\tan\theta_\mathrm{b} > 0$ indicate right-turn and left-turn, respectively. Figure 6(b) shows the torques produced by the scale factor, and two torques are roughly canceled out, i.e., moment equilibrium, $T_\mathrm{R} + T_\mathrm{N} = 0$. This allows us to reconstruct the terrain elevation and speed in metric units as shown in Figure 6(c). Note that the speed profile is physically meaningful, i.e., average speed of the mountain biking ranges between 2-10 m/s^2.

2. Previously, a reference physical quantity such as the height of an object [11], the baseline of stereo cameras, or an additional IMU sensor [37] have been used to estimate the true scale.



(a) Optimal active force (b) Linear trajectory (c) Angular trajectory

Fig. 7. (a) Equation (12) produces plausible active force and torque profile that produces a camera trajectory concerting with the video ((b) and (c)).

7 Solving Dynamical System

We solve the dynamical system in Equation (1) and (2) inspired by an optimal control theory [57]. From a first-person video, we detect the 2D feature points such as SIFT [35], $\widehat{\mathbf{x}}_{t,p} \in \mathbb{R}^2$, and compute the optimal sequence of active force and torque (control input) $\{\mathbf{u}_t\}$ that minimizes the reprojection error:

$$\begin{aligned} \underset{\{\mathbf{u}_t\}}{\text{minimize}} \quad & \sum_{t=0}^{T}\sum_{p=0}^{P} \delta_{t,p} \|\widehat{\mathbf{x}}_{t,p} - f_{\text{prj}}(\mathbf{P}(\mathbf{y}_t), \mathbf{X}_p)\|^2 \\ & + L_{\text{reg}}(\mathbf{u}_1, \cdots, \mathbf{u}_{T-1}) \\ \text{subject to} \quad & \mathbf{y}_{t+1} = f_{\text{dyn}}(\mathbf{y}_t, \mathbf{u}_t; \mathbf{G}, \alpha), \end{aligned} \tag{12}$$

where $\delta_{t,p}$ is the Kronecker delta function that produces 1 if $\mathbf{X}_p$ is visible from the t^{th} image, and 0 otherwise. L_{reg} is a regularization of the control input to enforce physical plausibility, e.g., smooth force application[3]. As a byproduct of the optimization, the scene 3D point and camera trajectory are reconstructed.

Solving Equation (12) involves with intractable optimization due to nonlinearity of projection f_{prj} and dynamics f_{dyn}. It requires a good initialization of the physical state $\{\mathbf{y}_t^0\}$ and control input $\{\mathbf{u}_t^0\}$ (Section 7.1), and a tractable algorithm (Section 7.2).

7.1 State Initialization via Structure from Motion

We initialize the physical state of the actor $\{\mathbf{y}_t^0\}$ by decoupling the reprojection from the dynamics. It is equivalent to structure from motion [19] that decomposes Equation (6) into the camera projection matrix and 3D point:

$$\underset{\{\mathbf{P}_i\},\{\mathbf{X}_p\}}{\text{minimize}} \quad \sum_{t=0}^{T}\sum_{p=0}^{P} \delta_{t,p} \|\widehat{\mathbf{x}}_{t,p} - f_{\text{prj}}(\mathbf{P}_t, \mathbf{X}_p)\|^2, \tag{13}$$

where $\mathbf{P}_i$ is parametrized by its rotation $\mathbf{R}_t$ and optical center $\mathbf{C}_t$. Given the reconstructed camera trajectory, the actor's state $\{\mathbf{y}_t^0\}$ can be computed given the gravity direction $\mathbf{G}$ and scale α.

3. $L_{\text{reg}}(\mathbf{u}_1, \cdots, \mathbf{u}_{T-1}) = \int_1^{T-1} \mathbf{u}\, dt$ is often used for a linear quadratic regulator.

The initialized physical state allows us to compute the corresponding control input $\{\mathbf{u}_t^0\}$ by solving Equation (5) for the linear and angular momentum, P_t and $\mathbf{L}_t$:

$$F_{\text{T}}^0 = \frac{P_{t+1} - P_t}{\Delta t} - F_{\text{R}} \tag{14}$$

$$T_{\text{Y}}^0 = \frac{\mathbf{L}_{t+1}^{\text{Y}} - \mathbf{L}_t^{\text{Y}}}{\Delta t} \tag{15}$$

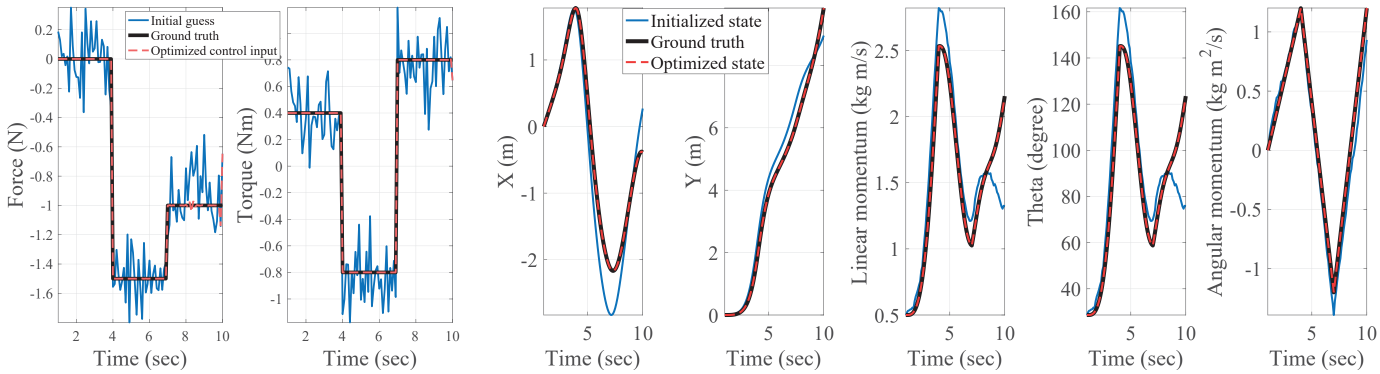
$$T_{\text{R}}^0 = \frac{\mathbf{L}_{t+1}^{\text{R}} - \mathbf{L}_t^{\text{R}}}{\Delta t} - \tau_{\text{R}} \tag{16}$$

where $\mathbf{L}_t^{\text{Y}}$ and $\mathbf{L}_t^{\text{R}}$ are the yaw and roll elements of the angular momentum. Since the initial control input $\{\mathbf{u}_t^0\}$ do not take into account the dynamics, it may not produce the corresponding physical state $\{\mathbf{y}_t^0\}$ that minimizes the reprojection error.

7.2 Camera Trajectory Following via Linearization

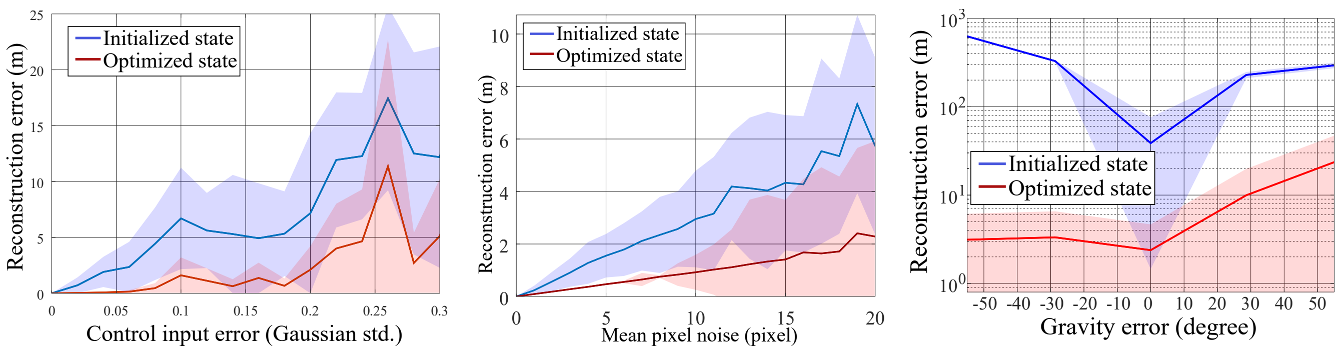
With the initialization, we compute the optimal active force and torque in Equation (12). We cast the problem as a nonlinear trajectory following that can be solved by an iterative linear quadratic regulator (iLQR) [32], [59]. We present a new iterative algorithm for the camera trajectory following designed to minimize reprojection error, which eventually solves for Equation (12). We iterate two processes: (1) compute the control policy by linearizing dynamics near a nominal trajectory, $\{\mathbf{y}_t\}$; (2) update the nominal trajectory based on the locally optimal control input, $\{\mathbf{u}_t\}$. The details of the linearization and algorithm can be found in Appendix B. A pseudo code is found in Algorithm 1 (Appendix B).

It is worth to highlight the main difference between Equation (12) and (13). Since both minimize the reprojection error, the kinematics of the reconstructed states are comparable. Figure 7(b) and 7(c) illustrated comparison of two reconstructed state (e.g., position, orientation, linear and angular velocity), i.e., $\{\mathbf{y}_t\} \approx \{\mathbf{y}_t^0\}$. However, the optimal active force and torque (control input) significantly differ from the initialization $\{\mathbf{u}_t\} \neq \{\mathbf{u}_t^0\}$ due to the decoupling of the dynamics in Equation (13) and physical plausibility applied through the regularization in Equation (12). Figure 7(a) shows the comparison between initial and optimized control input for the real mountain biking data where the initialization produces an implausible profile (e.g., noisy and discontinuous active force application).



(a) Active force and torque (b) Physical State

Fig. 8. We compute the optimal control (a) and its state (b) using Equation (12) that minimizes reprojection error for a synthetically generated motion. Given a camera trajectory, we initialize control input and then, compute the initial state (blue) using Equation (14).



(a) Active force and torque initialization error (b) Reprojection error (c) Gravity error

Fig. 9. We validate our algorithm using synthetic data. We add disturbance or noise in a form of control input, 2D projection, and gravity where the optimal control input produces accurate state trajectory.

8 Result

We evaluate our approach in terms of three key physical quantities: gravity, scale, and force/torque. Accuracy and error sensitivity (algorithmic robustness) are used for the evaluation metrics.

8.1 Validation via Synthetic Data

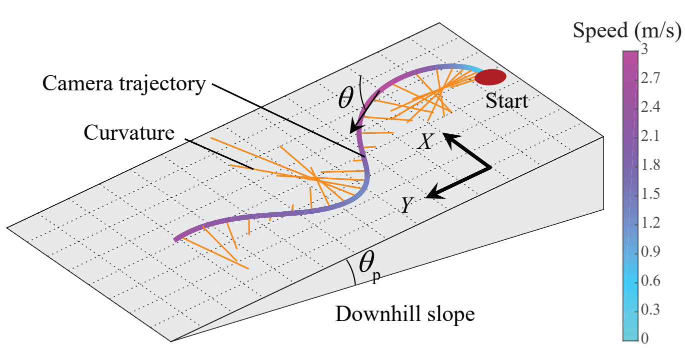
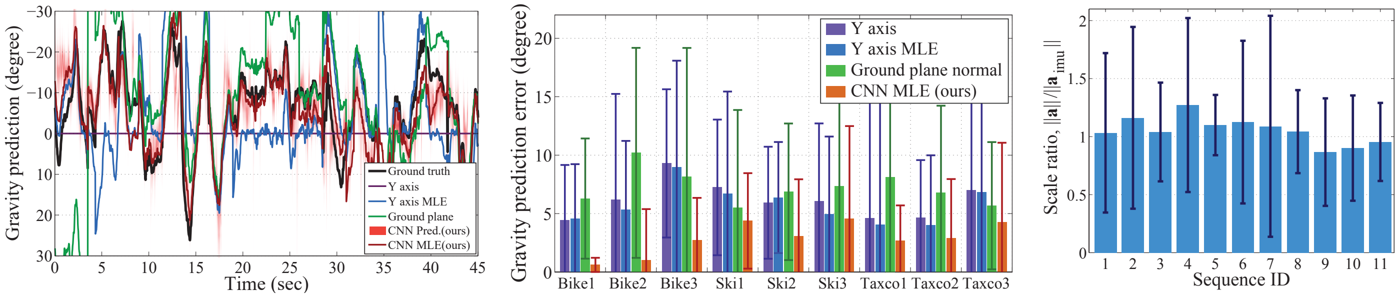


Fig. 10. We validate our method using synthetic data. The first-person motion on the downhill slop is generated where the color on the trajectory represents the speed and the yellow lines indicate direction and magnitude of curvature.

We analyze the algorithmic robustness on measurement errors using synthetic data, i.e., how much errors in scale, gravity, and state initialization can be tolerated by the optimization in Equation (12). We simulate an actor's motion sliding down in a downhill slope as shown in Figure 10. We generate random 1,200 3D points, $\mathbf{X}$, and 10 seconds of motion (300 frames). At least 40 random points per image in front of the camera are chosen to be visible, $\delta_{k,j}$ and the ground truth control input, $\{\mathbf{u}_t\}$ is designed to produce a natural S curve along the downhill. In Figure 10, the color on the trajectory represents the speed, and yellow lines indicate the direction and magnitude of its curvature at each time instant. For a demonstration purpose, we simplify the dynamics, i.e., restricting the motion to constant slope surface, θ_p: $\mathbf{y} = \begin{bmatrix} X & Y & P & \theta & \mathcal{J}\omega \end{bmatrix}^{\mathsf{T}}$ and $\mathbf{u} = \begin{bmatrix} F_{\mathrm{T}} & T_{\mathrm{Y}} \end{bmatrix}^{\mathsf{T}}$ where $Z = -Y\tan\theta_p$, θ and ω are the yaw angle and its velocity. Note that this simplification of state and control input is a special instance of Equation (5).

Robustness to initialization error of control input We recover the active thrust force and yaw torque (control input) and its physical states (location, linear momentum and angular momentum) from the erroneous initialization of $\{\mathbf{u}_t^0\}$ and $\{\mathbf{y}_t^0\}$. We add Gaussian noise on the ground truth trajectory to compute the control input force, which is used for initialization. Figure 8 illustrates the estimated trajectory of state and control input. The initialization of the control produces the noisy trajectory (blue line). The optimization in Equation (12) finds the optimal trajectory (red dotted line) converging to the ground truth trajectory (black line). The noisy control (Figure 8(a)) is also optimized, which produces plausible and smooth control profile aligned with the ground truth. Figure 9(a) shows state reconstruction error as varying the control error where we compare the trajectories



(a) Gravity estimation (b) Gravity prediction comparison (c) Scale prediction

Fig. 11. (a) We compare our gravity estimation with three baseline algorithms (see the description of the baseline algorithm in Section 8.2. The red heatmap indicates the likelihood at each time instant. (b) We measure error across different scenes. (c) We recover physical scale and compare with IMU in terms of linear acceleration. Our method correctly estimate the scale (perfect recovery if 1; median 1.0287 with 0.6186 standard deviation).

of initialized and optimized states. The optimized state reduces reconstruction error: 0.2 N std. produces less than 2 m error.

Robustness to reprojection error Our system relies on structure from motion to initialize the state and control input. The reprojection error after bundle adjustment varies 0.5 to 3 pixels in practice. In Figure 9(b), we show the robustness with respect to reprojection error. Given ground truth 2D projection, we add Gaussian noise that affects the Jacobian in Equation (20). Our optimized control input produces accurate state prediction in the presence of significant pixel noise (1 m reconstruction error at 10 mean pixel noise).

Robustness to gravity error Our control force and torque computation uses the gravity estimate as an input. The gravity prediction could be erroneous when the visual semantics is consistently confusing across time as shown in the right image of Figure 4. In Figure 9(c), we illustrate the reconstruction error as varying the gravity error. Our algorithm is resilient to the gravity offset, i.e., -50 degree with a few meters of reconstruction. The reconstruction error is not symmetric because the sign of the slope θ_p changes as the gravity error increases.

8.2 Quantitative Evaluation

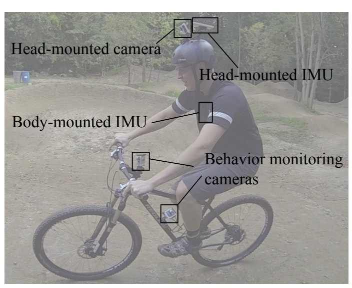


Fig. 12. Controlled experiments with an experienced mountain biker.

We quantitatively evaluate our algorithm with a controlled experiment conducted by an experienced mountain biker. The biker wore head-mounted camera and inertial measurement unit (IMU) as shown in Figure 12. Additionally, two IMUs are attached on his torso near the center of his body mass to measure disparity between head and body motion. Two more cameras are also attached on the bike to monitor his control input, i.e., pedaling and braking activity. Our evaluations are performed to verify our method in three criteria: gravity prediction, scale recovery, and active force and torque estimation.

Gravity prediction We train a CNN model for each activity using approximately 50,000 images with gravity annotations (nearly 5 hour long sequence)[4]. We compare our prediction using CNN and reconstructed camera orientation with three baseline methods: a) Y axis: prediction by the image Y axis as a camera is often oriented upright; b) Y axis MLE: prediction by a) consolidated by the reconstructed camera orientation; c) ground plane normal. The ground plane is estimated by fitting a plane with RANSAC on the sparse point cloud. Figure 11(a) shows a comparison with baseline algorithms where our method produces median error 2.7 degree with 3.64 standard deviation (mean: 4.40 degree). Note that we do not compare our final MAP estimate for fair comparison. We also test our method on manually annotated data in Figure 11(b) where our method consistently outperforms others significantly ($\times 2 \sim \times 10$). Note that only biking sequences are used for the training data while Bike 1, 2, and 3 were not included in the training dataset. Table 2 (Appendix A) summarizes the gravity prediction for various activities, e.g., Moutain biking, skiing, urban biking (Taxco).

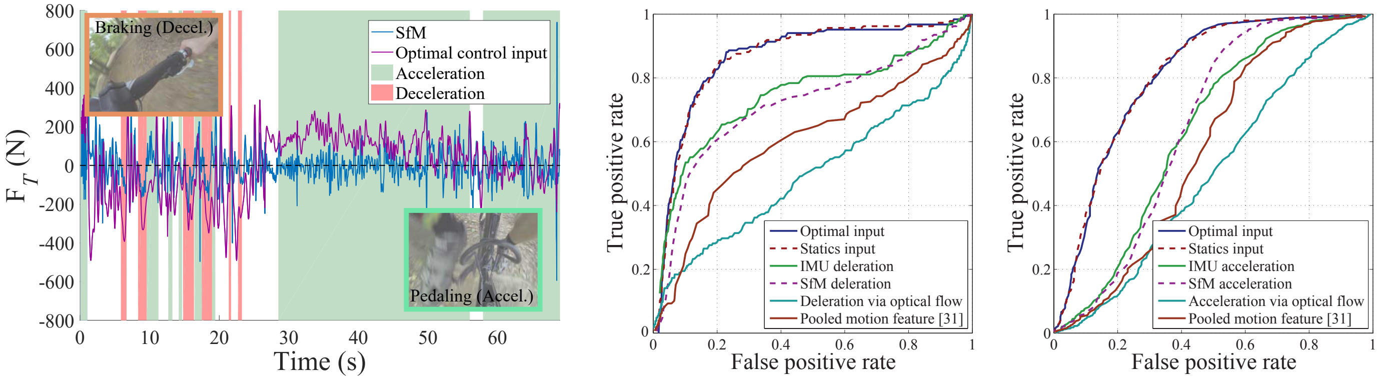
Scale recovery We recover the scale factor and compare the magnitude of linear acceleration with IMU, i.e., $\|\mathbf{a}\|/\|\mathbf{a}_{\rm m}\|$ where $\mathbf{a}$ and $\mathbf{a}_{\rm imu}$ are acceleration of ours and IMU, respectively. Note that IMU data is noisier than our estimation but the ratio remains approximately 1 (head: 1.0278 median, 1.1626 mean, 0.6186 std.; body: 0.9999 median, 1.1600 mean, 0.7739 std.). We recover scale factors for 11 different sequences each ranges between 1 mins to 15 mins as shown in Figure 11(c). This results in overall 1.0188 median, 1.1613 mean, and 0.7003 std.

Active force estimation We identify the moment that thrust force (pedaling and braking) is applied[5]. We use a thresholding binary classifier, $\xi^+(t)$ and $\xi^-(t)$ to detect pedaling and braking, respectively: $\xi^+(t) = 1$ if $\int_{t-1}^{t} F_{\rm T}(t){\rm d}t > \epsilon_T$, and 0 otherwise; $\xi^-(t) = 1$ if $\int_{t-1}^{t} F_{\rm T}(t){\rm d}t < -\epsilon_T$, and 0 otherwise[6]. Figure 13(a) shows active force profile and ground truth manually annotated from the videos of behavior monitoring cameras as shown in Figure 12. Our active force profile accords with the ground truth, i.e., pedaling when $F_{\rm T} > 0$ and braking when $F_{\rm T} < 0$. We compare it with the active thrust force estimation using structure from motion which is not indicative of pedaling and braking actions. In Figure 13(b) and 13(c), we compare our method with net acceleration measured by IMU and structure from motion. We also compare against optical flow to measure acceleration that is often use for egocentric activity recognition

4. The pre-trained model is publicly available: http://www-users.cs.umn.edu/∼hspark/ffm.html

5. Active force and torque cannot be measured by IMU because the measured acceleration is due to net force and torque not input. This requires force/torque sensors attached human bodies that measures muscle tension.

6. A sophisticated classifier such as recurrent neural networks can be a complementary approach when supervision is available.



(a) Accel/decel detection (b) Acceleration (c) Deceleration

Fig. 13. (a) We compare active forces computed by optimal control input (ours) and structure from motion with the ground truth braking and pedaling actions. The positive thrust force F_T from our approach is highly correlated with pedaling and negative thrust force with braking. (b) and (c) Our method outperforms optical flow based representation including [53] with a large margin.

tasks [28], [52]. Also we compare with Pooled Motion Feature representation [53], which requires a pre-trained model. Our active force identification outperforms other baseline methods that do not take into account active force decomposition. This verifies that a trivial extension by attaching IMU on camera is not sufficient enough to estimate the active force applied by the actor—the measured acceleration needs to be decomposed.

Active torque estimation We compare the estimated angular velocity with measurements from gyroscope in Figure 14(a). Note that the velocity computation by differentiating the reconstructed camera trajectory does not directly apply as different framerate between IMU and camera and noisy reconstruction. The optimally estimated active force and torque generate plausible angular velocity profile. Table 1 summarizes error of angular velocity measured by 11 different scenes. The correlation is also measured, which produces 0.87 mean correlation.

	1	2	3	4	5	6	7	8	9	10	11
Mean(rad/sec)	0.25	0.31	0.27	0.31	0.27	0.26	0.41	0.29	0.30	0.30	0.40
Med. (rad/sec)	0.18	0.30	0.17	0.27	0.26	0.22	0.36	0.23	0.22	0.24	0.36
Std. (rad/sec)	0.24	0.20	0.26	0.23	0.19	0.19	0.32	0.23	0.27	0.26	0.31
Corr.	0.91	0.94	0.90	0.88	0.88	0.61	0.82	0.83	0.90	0.86	0.86

TABLE 1
Angular velocity comparison with gyroscope. Med.: median, Std.: standard deviation, Corr: correlation (perfect if 1)

8.3 Qualitative Evaluation

We apply our method on real world data downloaded from YouTube. 5 different types of scenes are processed: 1) mountain biking (1-10 m/s); 2) Flying: wingsuit jump (25-50 m/s) and speedflying with parachute (9-40 m/s); 3) jetskiing at Canyon (4-20 m/s); 4) glade skiing (5-12 m/s); 5) Taxco urban downhill biking (5-15 m/s). Figure 15 illustrate estimated gravity direction, physical scale of force and velocity, and active force and torque. Also passive components such as air drag, pitch torque, and normal force are shown. Thrust force is applied when climbing up the hill in Biking or when accelerating in Jetskiing. For Skiing, periodic lateral forces and roll moments are observed as the actor was banking frequently. For flying case[7], strong air drag force and lifting forces are observed. Also unstable angular momentum along the roll axis comparing to other axes is observed, which requires skillful body control to balance left and right wings. We assume that all videos have the same intrinsic parameter (fisheye distortion [13], ω=0.001619; focal length, f_x=547.55, f_y=535.48; principal coordinates, p_x=640, p_y=360) and image resolution (1280×720), which we pre-calibrate with GoPro Hero 3 Black edition at the $1280{\times}720$ resolution.

9 Limitations

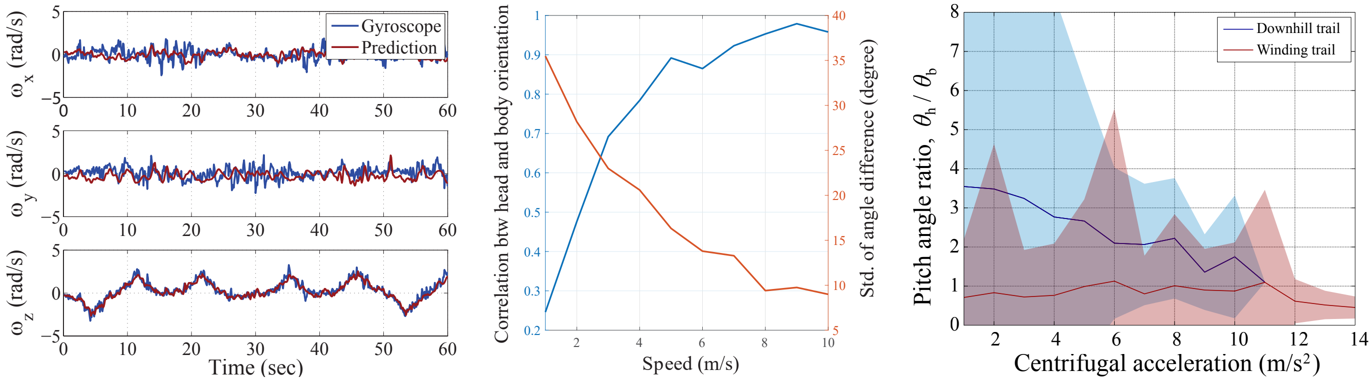
We make a few assumptions that enable us to map the first-person visual scene to the actor's physical state. Albeit valid in many practical cases, the assumptions do not always hold, which produces a degenerate solution. In this section, we discuss the limitations of the model.

Physical scale recovery There exists a degenerate case of the physical scale recovery in Equation (11). The scale factor $\alpha = 0$ if $\theta_\mathrm{b} = 0$, i.e., zero centripedal acceleration. This occurs when the camera wearer never changes a direction (i.e., making a banked turn) in the course of the entire first-person video. This is unlikely for sport activities[8]. A potential solution for such videos is to leverage the gravitational acceleration along the forward motion, i.e., projectile motion of jumping and dominant forward acceleration along the downhill slope.

Camera placement In Assumption 1, we approximate the center of mass and its orientation with the 3D camera pose. While it allows a key simplification of the projection model in Equation (6), it can produce an estimation bias. *Angular bias:* We found that the angular approximation using the camera is valid when the actor undergoes rapid movement where the head and body orientation becomes highly correlated. We empirically validate this approximation by measuring the correlation between body and head orientation. Strong correlation is observed at high speed (corr.: 0.96; std.: 9 degree at 10 m/s) as shown in Figure 14(b). Also a similar correlation is observed when the camera wearer undergoes high centrifugal acceleration as shown in Figure 14(c),

7. Unfortunately, the gravity direction cannot properly estimated as it was even challenging to a human annotator. Instead, we manually find frames that contain the horizon to estimate the gravity direction.

8. No such instance was found in our collection of YouTube first-person videos (156 sequences of more than 10 hours of videos).



(a) Angular velocity (b) Body-head correlation (c) Body-centripetal correlation

Fig. 14. (a) We compare our estimation with a gyroscope attached to the camera. Our estimation via active force and torque produces plausible angular velocity profile that accords with the gyroscope measurements. (b) The body orientation can differ from the head orientation by 60 degree in general. However, as the actor experiences higher speed, the body and head orientations are strong correlated. This correlation validates the approximation of center of mass using the first-person camera pose. (c) Such trend appears when the actor experiences higher centripetal acceleration (smaller curvature and faster instantaneous velocity).

which validates physical scale recovery from a banked turn in Section 6.2. Nonetheless, it is possible that the estimation could be inaccurate due to the actor's head movements independent to the egomotion (e.g., distraction). *Positional bias:* According to D'Alembert's principle of inertial forces, the angular momentum is independent of the location of center of mass, i.e., the active torques, $T_{\rm Y}$ and $T_{\rm R}$, remain constant. However, the torques induced by forces such as centripedal force change due to the length of levers, which can produce inaccurate angular momentum computation. This limitation may be addressed by a two-link inverted pendulum model.

Model expressibility We simplify the dynamics by limiting the active force an torque along thrust, yaw, and roll directions, which are the dominant active components. Such force and torque decomposition may not be valid for all activities. For instance, an active pitch torque $T_{\rm P}$ may play a roll for an acrobatic motion of biking and an intended stall motion of a wingsuit flying activity.

10 Discussion

In this paper, we present a method to reconstruct physical sensation of a first person video. We recover three ingredients for the physical sensations: gravity direction, physical scale, and active force and torque. The gravity direction is computed by leveraging a convolutional neural network integrated with the reconstructed 3D camera orientations. We recover the physical scale by using a torque equilibrium relationship along the roll axis at a bank turn. Active and passive components are modeled using rigid body dynamics which is integrated into the 3D reconstruction pipeline. We quantitatively evaluate our method with controlled experiments where our method outperforms other baseline algorithms with a large margin ($\times 2 \sim \times 10$) and apply our method on real world data of various activities such as biking, skiing, flying, jetskiing, and urban bike racing.

The main computational bottle neck of our system is the initialization by structure from motion. In our experiments, 1 minutes of video (1,800 images) took more than 5 hours accelerated by multicore CPUs (64× Intel Xeon 7500). The rest of computations (gravity with nVidia TitanX, scale, active force) took less than 5 minutes.

This paper opens up a new opportunity to understand and analyze human activities using a first-person video in terms of *controls*. While sporting activities are of main interest of this paper, enabling this approach for daily activities such as cooking, exercising, and social interactions will bring out new compelling applications towards computational wearable technology.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015.
[2] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, 2015.
[3] I. Arev, H. S. Park, Y. Sheikh, J. K. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. *SIGGRAPH*, 2014.
[4] R. Baillargeon. How do infants learn about the physical world? *Current Directions in Psychological Science*, 1994.
[5] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, 2014.
[6] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. First person action-object detection with egonet. In *arXiv:1603.04908*, 2016.
[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000.
[8] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998.
[9] M. A. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics. In *CVPR*, 2007.
[10] K. Choo and D. J. Fleety. People tracking using hybrid monte carlo filtering. In *ICCV*, 2001.
[11] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 2000.
[12] A. Dal Monte, L. M. Leonardi, C. Menchinelli, and C. Marini. A new bicycle design based on biomechanics and advanced technology. *International Journal of Sport Biomechanics*, 1987.
[13] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 2001.
[14] I. S. Dhillon and S. Sra. Modeling data using directional distributions. Technical Report TR-03-06, The University of Texas at Austin, 2003.
[15] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011.
[16] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, 2012.
[17] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IROS*, 2013.
[18] M. R. Greene and A. Oliva. Recognition of natural scenes from global properties: Seeing the forest without representing the trees. *Cognitive Psychology*, 2009.
[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

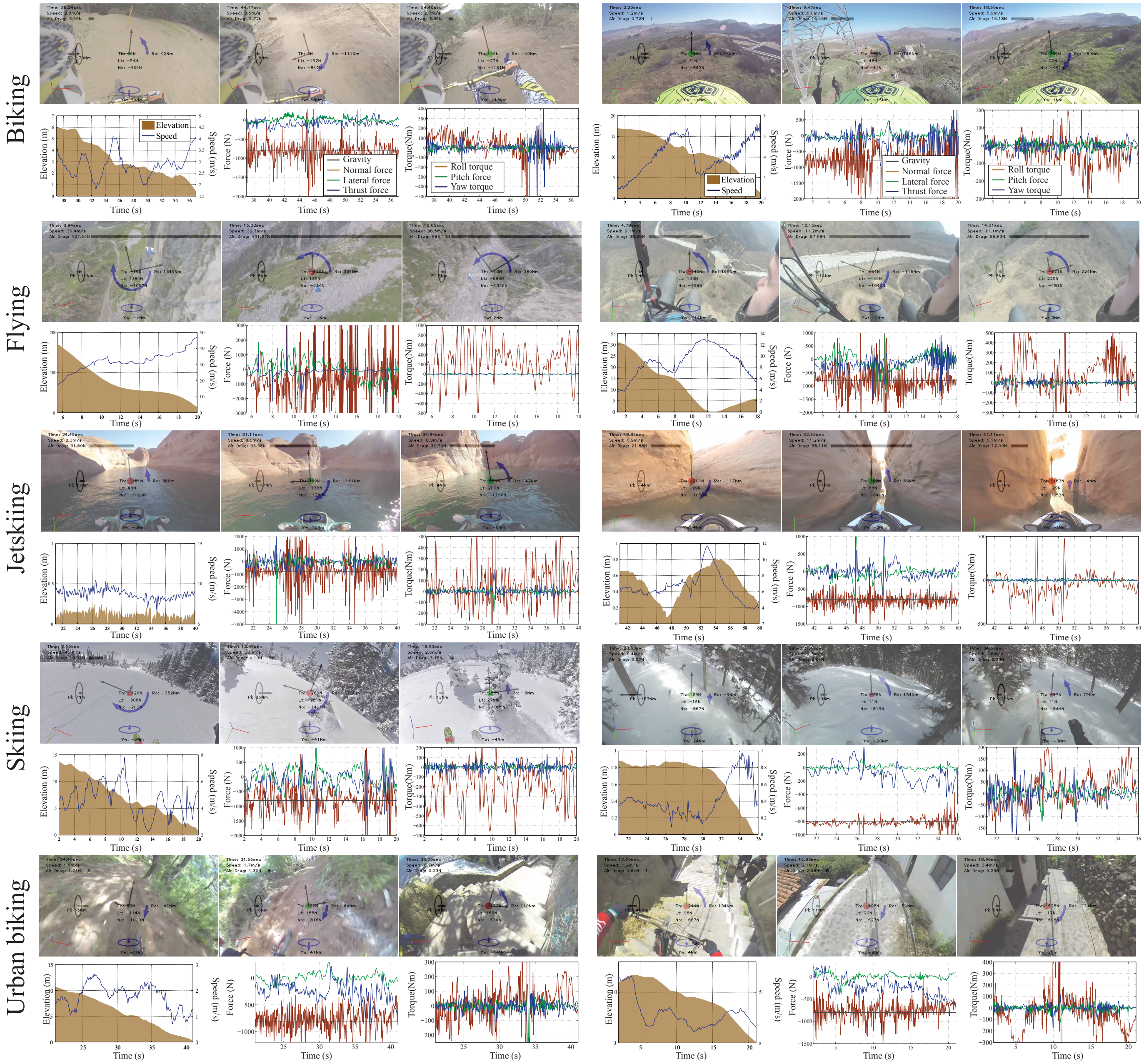


Fig. 15. We compute gravity direction, physical scale factor, and active force and torque from a first person video. For each sequence, the top row shows image superimposed with speed, gravity, forces, and torque. Full trajectories of such physcial quantities are illustrated in the next row.

[21] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *NIPS*, 2000.
[22] S. K. I. Akhter, Y. Sheikh and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *TPAMI*, 2011.
[23] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *ACCV*, 2014.
[24] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015.
[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
[26] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophycics*, 1973.
[27] T. Kanade and M. Hebert. First person vision. In *IEEE*, 2012.
[28] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011.
[29] J. Kopf, M. Cohen, and R. Szeliski. First person hyperlapse videos. *SIGGRAPH*, 2014.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
[31] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.
[32] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *International Conference on Informatics in Control, Automation and Robotics*, 2004.
[33] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *ICCV*, 2013.
[34] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
[35] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
[36] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *TPAMI*, 1993.
[37] F. M. Mirzaei and S. I. Roumeliotis. A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. *TRO*, 2008.
[38] J. K. Moore, J. D. G. Kooijman, M. Hubbard, and A. L. Schwab.

A method for estimating physical properties of a combined bicycle and rider. In *International Design Engineering Technical Conferences Computers and Information in Engineering Conference*, 2009.
[39] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 2001.
[40] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. In *CVPR*, 2016.
[41] H. S. Park, J.-J. Hwang, and J. Shi. Force from motion: Decoding physical sensation in a first person video. In *CVPR*, 2016.
[42] H. S. Park, E. Jain, and Y. Shiekh. 3D social saliency from head-mounted cameras. In *NIPS*, 2012.
[43] H. S. Park and Y. Sheikh. 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. In *ICCV*, 2011.
[44] H. S. Park and J. Shi. Social saliency prediction. In *CVPR*, 2015.
[45] H. Pirsiavash and D. Ramanan. Recognizing activities of daily living in first-person camera views. In *CVPR*, 2012.
[46] G. Pusiol, L. Soriano, L. Fei-Fei, and M. C. Frank. Discovering the signatures of joint attention in child-caregiver interaction. In *CogSci*, 2014.
[47] F. A. Raviteja Vemulapalli and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014.
[48] J. M. Rehg, G. D. Abowd, A. Rozga, M. Romero, M. A. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, H. Rao, J. C. Kim, L. L. Presti, J. Zhang, D. Lantsman, J. Bidwell, and Z. Ye. Decoding childrens social behavior. In *CVPR*, 2013.
[49] G. Robson and R. D'Andrea. Longitudinal stability analysis of a jet-powered wingsuit. In *AIAA Atmospheric Flight Mechanics Conference*, 2010.
[50] G. Rogez, J. Supancic, and D. Ramanan. First-person pose recognition using egocentric workspaces. In *CVPR*, 2015.
[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.
[52] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me. In *CVPR*, 2013.
[53] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *CVPR*, 2015.
[54] S. Shin, Y. Ahn, J. Choi, and S. Han. Design of a framework for interoperable motion effects for 4d theaters using human-centered motion data. In *International Conference on Advances in Computer Entertainment Technology*, 2010.
[55] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000.
[56] K. K. Singh, K. Fatahalian, and A. A. Efros. Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *WACV*, 2016.
[57] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, 1998.
[58] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single image. In *CVPR*, 2000.
[59] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference*, 2005.
[60] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *TPAMI*, 2008.
[61] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *CVPR*, 2006.
[62] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3-d human body tracking. *CVIU*, 2006.
[63] J. Valmadre, Y. Zhu, S. Sridharan, and S. Lucey. Efficient articulated trajectory reconstruction using dynamic programming and filters. In *ECCV*, 2012.
[64] M. Vondrak, L. Sigal, and O. Jenkins. Physical simulation for probabilistic motion tracking. In *CVPR*, 2008.
[65] X. Wei and J. Chai. Videomocap: Modeling physically realistic human motion from monocular video sequences. *SIGGRAPH*, 2010.
[66] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan. Principles of sensorimotor learning. *Nature Reviews Neuroscience*, 2011.
[67] C. R. Wren and A. Pentland. Dynamic models of human motion. In *IEEE Face and Gesture*, 1998.
[68] B. Xiong and K. Grauman. Detecting snap points in egocentric video with a web photo prior. In *ECCV*, 2014.
[69] J. Yan and M. Pollefey. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *TPAMI*, 2008.
[70] Z. Ye, Y. Li, Y. Liu, C. Bridges, A. Rozga, and J. M. Rehg. Detecting bids for eye contact using a wearable camera. In *FG*, 2015.
[71] R. Yonetani, K. M. Kitani, and Y. Sato. Ego-surfing first person videos. In *CVPR*, 2015.

Hyun Soo Park Hyun Soo Park is an assistant professor in the Department of Computer Science and Engineering at the University of Minnesota. He is interested in understanding human visual sensorimotor behaviors from first person cameras. Prior to joining the UMN, he was a postdoctoral fellow in the GRASP Lab at the University of Pennsylvania, and earned his Ph.D. from Carnegie Mellon University.

Jianbo Shi Jianbo Shi studied Computer Science and Mathematics as an undergraduate at Cornell University where he received his B.A. in 1994. He received his Ph.D. degree in Computer Science from University of California at Berkeley in 1998. He joined The Robotics Institute at Carnegie Mellon University in 1999 as a research faculty, and in 2003 University of Pennsylvania where he is currently a Professor of Computer and Information Science. In 2007, he was awarded the Longuet-Higgins Prize for his work on Normalized Cuts. His current research focuses on first person vision, human behavior analysis and image recognition-segmentation. His other research interests include image/video retrieval, 3D vision, and vision based desktop computing. His long-term interests center around a broader area of machine intelligence, he wishes to develop a ”visual thinking” module that allows computers not only to understand the environment around us, but also to achieve cognitive abilities such as machine memory and learning.

Appendix A
Gravity Training Data

We learn motion and visual cues of the gravitational field, which requires a gravity annotated training data. We employ two methods to annotate a large scale first-person video data.

Controlled data with IMU measurements We rigidly attach a 6 DOF inertial measurement unit (IMU) sensor to a first person camera and extracted synchronized IMU and images after visual-inertial calibration [17]. The gravity vector is computed using acceleration and gyroscope measurements where we associate it with image and its spatial rotation, i.e., $\mathbf{G} \leftrightarrow \{\mathcal{I}, \mathbf{R}\}$. This gravity annotation scheme is fully automatic, which is scalable.

Uncontrolled data without IMU measurements We augment the gravity annotation data by using the images extracted from first-person videos in Internet. Unlike the controlled data, the ground truth gravity vector is not available for these images. Training a CNN model requires a large amount of manual efforts. Instead of labeling each image, we develop a new efficient annotation scheme that can geometrically propagate a few manual annotations across the entire images using 3D reconstruction of the camera poses. A key idea is that there exists a global 3D gravity vector $\mathbf{G}(\phi_1, \phi_2)$ that is consistent across the entire video, i.e., the estimated 3D gravity vector from the annotations of a few images can be projected to the rest images for the propagation of the gravity annotation.

Given a set of a few manual annotations in 2D, $(\mathbf{I}, \theta_g)$, we reconstruct the global gravity direction in 3D by minimizing reprojection error of image orientation:

$$\mathbf{G}(\phi_1^*, \phi_2^*) = \underset{\mathbf{G}(\phi_1,\phi_2)}{\operatorname{argmin}} \sum_{k=1}^{K} \left(\theta_g^k - \tan^{-1} \frac{(\mathbf{R}_k^x)^\mathsf{T}\mathbf{G}}{(\mathbf{R}_k^y)^\mathsf{T}\mathbf{G}} \right)^2 \quad (17)$$

where $\theta_g^k \in [-\pi/2, \pi/2]$, $\mathbf{R}_k^x$, and $\mathbf{R}_k^y$ are the k^{th} annotated 2D orientation, first and second rows of the reconstructed rotation matrix, $\mathbf{R}_k$. Equation (17) is the maximum likelihood estimate of the 3D gravity given annotated images. We solve this by finding global minimum via enumerating discretized $\phi_1 \in [\ 0 \quad \pi\ ]$ and $\phi_2 \in [\ 0 \quad 2\pi\ ]$ and refine the solution using a gradient decent optimization. Note that $\mathbf{G}$ can be ambiguous if $\{\mathbf{R}_k\}_{k=1}^K$ undergoes zero yaw angular displacement (rotation about Y axis of the camera) as shown in Figure 5.

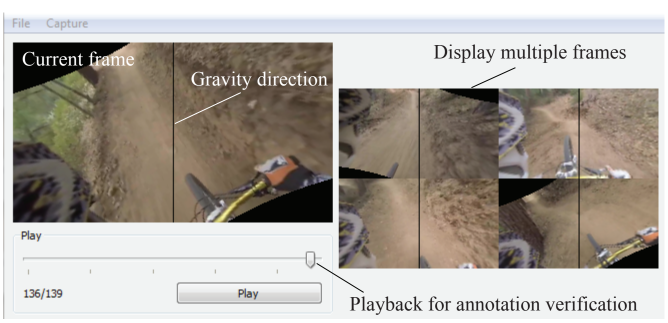


Fig. 16. We annotate a gravity direction using our tool that allows us to annotate 100 frames at once.

To facilitate the manual annotations, we develop an interactive gravity annotation software that can align the global gravity with multiple images as shown in Figure 16. A user control (ϕ_1, ϕ_2), which changes the orientations of multiple images, simultaneously, computed by structure from motion. We visualize the lines passing a vertical vanishing point to align with a vertical scene structure such as trees. This allows precise gravity annotation ($< 5^\circ$ in 3D) for an entire video at once. In practice, we annotate a few images (10-30) with sufficient yaw angular displacement where $\mathbf{G}$ can be uniquely computed, and then, propagate it to the rest of images (2,000-5,000). This annotation method allows us to label more than 300,000 frames with less than 2,000 image labeling, i.e., a 5 min video (9,000 frames) in 3 mins.

Details of Training We augment the training data by rotating the image about the principal point, $\mathcal{I}^a = \mathcal{I}(\mathbf{K}\mathbf{M}_{\theta_a}\mathbf{K}^{-1}\mathbf{x})$ where $\mathbf{M}_{\theta_a} \in \mathrm{SO}(3)$ is in-plane rotation (rotation about z axis) with θ_a angle. This data augmentation allows handling large roll angle displacement of first person video and balance the distribution of the gravity labels. We also augment the data by flipping horizontally. We implement the gravity model (AlexNet [30]) in Caffe [25] on a single GPU (nVidia GTX) for 50K iterations. The Table 2 summarizes the results of gravity prediction. The pre-trained model is publicly available: http://www-users.cs.umn.edu/~hspark/ffm.html

Appendix B
iLQR for Solving Dynamical System

We formulate the estimation of the active force and torque from first-person images using iLQR, i.e., camera trajectory following. We linearize the dynamics in Equation (1) and projection in Equation (2) to compute the control policy gradient.

Linearization of dynamics Let $\delta\mathbf{y}_t$ and $\delta\mathbf{u}_t$ be the deviation from the nominal state and control input, respectively, i.e., $\mathbf{y}_t^{i+1} = \mathbf{y}_t^i + \delta\mathbf{y}_t$ and $\mathbf{u}_t^{i+1} = \mathbf{u}_t^i + \delta\mathbf{u}_t$ where the superscript represents the iteration number. We linearize the state dynamics in Equation (5) using the first order Taylor expansion:

$$\mathbf{y}_{t+1}^{i+1} - \mathbf{y}_{t+1}^{i} = \delta\mathbf{y}_{t+1} = \mathbf{A}_t\delta\mathbf{y}_t + \mathbf{B}_t\delta\mathbf{u}_t, \quad (18)$$

where the analytic form of $\mathbf{A}_t = \left.\frac{\partial f_{\text{dyn}}}{\partial \mathbf{y}}\right|_{\mathbf{y}_t^i} \in \mathbb{R}^{11\times 11}$ and $\mathbf{B}_t = \left.\frac{\partial f_{\text{dyn}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_t^i} \in \mathbb{R}^{11\times 3}$ can be found in the Appendix C.

Linearization of projection We approximate the 2D projection, $\widehat{\mathbf{x}}$, near nominal state trajectory as follow:

$$f_{\text{prj}}(\mathbf{P}(\mathbf{y} + \delta\mathbf{y}), \mathbf{X}) \approx f_{\text{prj}}(\mathbf{P}(\mathbf{y}), \mathbf{X}) + \frac{\partial f_{\text{prj}}}{\partial \mathbf{y}}\delta\mathbf{y}, \quad (19)$$

where $\frac{\partial f_{\text{prj}}}{\partial \mathbf{y}}$ is the Jacobian of the projection where its analytic form can be found in the Appendix C.

This allows linearizing the reprojection error in Equation (12):

$$\sum_{t=0}^{T}\sum_{p=0}^{P} \delta_{t,p}\|\widehat{\mathbf{x}}_{t,p} - \mathbf{x}_{t,p}\|^2 \approx \sum_{t=1}^{T} \|\mathbf{e}_t - \mathbf{D}_t\delta\mathbf{y}_t\|^2, \quad (20)$$

where

$$\mathbf{e} = \begin{bmatrix} \delta_{t,1}(\widehat{\mathbf{x}}_{t,1} - \mathbf{x}_{t,1}) \\ \vdots \\ \delta_{t,P}(\widehat{\mathbf{x}}_{t,P} - \mathbf{x}_{t,P}) \end{bmatrix}, \ \mathbf{D}_t = \begin{bmatrix} \delta_{t,1}\frac{\partial \mathbf{x}_{t,1}}{\partial \mathbf{y}} \\ \vdots \\ \delta_{t,P}\frac{\partial \mathbf{x}_{t,P}}{\partial \mathbf{y}} \end{bmatrix},$$

where $\mathbf{x}_{t,p} = f_{\text{prj}}(\mathbf{P}(\mathbf{y}_t), \mathbf{X}_p)$.

Recursive cost function By combining Equation (12) and (20), we can derive a recursive cost-to-go function:

$$J_t = \delta\widetilde{\mathbf{y}}_t^\mathsf{T}\mathbf{Q}_t\delta\widetilde{\mathbf{y}}_t + \delta\widetilde{\mathbf{u}}_t^\mathsf{T}\mathbf{W}\delta\widetilde{\mathbf{u}}_t + \min_{\delta\mathbf{u}_t} J_{t+1}, \quad (21)$$

	Bike 1			Bike 2			Bike 3			Bike IMU			Ski 1		
	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.
Y axis	5.62	4.44	4.72	8.10	6.18	9.06	10.15	9.29	6.34	16.02	13.11	10.88	8.31	7.24	5.80
Y axis MLE	5.92	4.57	4.66	6.08	5.31	5.91	10.68	8.97	9.11	15.83	12.28	11.21	10.09	6.72	8.72
Ground plane	7.45	6.28	5.14	12.69	10.20	8.99	11.31	8.16	11.01	11.98	10.24	9.03	8.27	5.50	8.36
CNN MLE (ours)	**0.76**	**0.61**	0.60	**2.53**	**1.00**	4.38	**4.40**	**2.70**	3.64	**11.21**	**9.11**	8.18	**5.17**	**4.37**	4.08

	Ski 2			Ski 3			Taxco 1			Taxco 2			Taxco 3		
	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.	Mean	Med.	Std.
Y axis	8.11	7.37	6.94	6.86	5.93	4.79	8.00	4.62	13.10	5.77	4.66	4.92	9.66	7.00	8.84
Y axis MLE	7.80	6.54	6.28	7.00	6.37	4.75	6.90	4.06	12.73	5.94	4.01	5.97	10.41	6.83	10.85
Ground plane	7.36	6.90	5.17	7.87	6.86	5.84	10.44	8.13	13.04	8.07	6.79	7.44	7.09	5.67	5.44
CNN MLE (ours)	**4.97**	**2.59**	11.17	**4.53**	**3.05**	4.88	**3.37**	**2.68**	3.02	**4.60**	**2.89**	5.06	**5.86**	**4.26**	6.80

TABLE 2
Gravity prediction error (degree). Med.: median, Std.: standard deviation

where J_t is accumulated cost at the t^{th} time instance and $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{u}}$ are the homogeneous representation of $\mathbf{y}$ and $\mathbf{u}$, respectively. The first two terms in Equation (21) measures the reprojection error and regularization loss where

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{D}_t^\mathsf{T}\mathbf{D}_t & -\mathbf{D}_t^\mathsf{T}\mathbf{e}_t \\ -\mathbf{e}_t^\mathsf{T}\mathbf{D}_t & \mathbf{e}^\mathsf{T}\mathbf{e} \end{bmatrix}, \quad \mathbf{W}_t = \begin{bmatrix} \mathbf{I}_2 & -\mathbf{d}_t \\ -\mathbf{d}_t^\mathsf{T} & \mathbf{d}_t^\mathsf{T}\mathbf{d}_t \end{bmatrix},$$

where $\mathbf{d}_t = \mathbf{u}_t^1 - \mathbf{u}_t^i$. The regularization via $\mathbf{W}_t$ prevents $\mathbf{u}_t^i$ deviates too much from the nominal control input $\mathbf{u}_t^1$.

The accumulated cost, J_t in Equation (21) can be rewritten in a closed form based on the current nominal state, $\mathbf{y}_t$, i.e., $J_t = \delta\widetilde{\mathbf{y}}_t^\mathsf{T}\mathbf{S}_t\delta\widetilde{\mathbf{y}}_t$ where

$$\begin{aligned} \mathbf{S}_t &= \mathbf{Q}_t + \mathbf{H}_t^\mathsf{T}\mathbf{W}\mathbf{H}_t + \left(\widetilde{\mathbf{A}}_t + \widetilde{\mathbf{B}}_t\mathbf{H}_t\right)^\mathsf{T}\mathbf{S}_{t+1}\left(\widetilde{\mathbf{A}}_t + \widetilde{\mathbf{B}}_t\mathbf{H}_t\right) \\ \mathbf{H}_t &= -\left(\mathbf{W} + \widetilde{\mathbf{B}}_t^\mathsf{T}\mathbf{S}_{t+1}\widetilde{\mathbf{B}}_t\right)^{-1}\widetilde{\mathbf{B}}_t^\mathsf{T}\mathbf{S}_{t+1}\widetilde{\mathbf{A}}_t, \end{aligned} \tag{22}$$

where $\widetilde{\cdot}$ represents the transition matrix for the homogeneous representation, $\widetilde{\mathbf{y}}_t$, i.e., $\delta\widetilde{\mathbf{y}}_{t+1} = \widetilde{\mathbf{A}}_t\delta\widetilde{\mathbf{y}}_t + \widetilde{\mathbf{B}}_t\delta\mathbf{u}_t$.

Algorithm 1 Camera trajectory following

Input: $\mathbf{X}_j \leftrightarrow \mathbf{x}_{t,j}$ and $\{\mathbf{C}_t, \mathbf{R}_t\}_{t=1}^T$
$\mathbf{y}_1 = \begin{bmatrix} \mathbf{C}_1^\mathsf{T} & P_1 \mid \mathbf{q}_1^\mathsf{T} & \mathbf{L}_1^\mathsf{T} \end{bmatrix}^\mathsf{T}$
$\delta\mathbf{y}_1 = \mathbf{0}^\mathsf{T}$
for $t = 1 : T-1$ **do**
  Compute initialization $\mathbf{u}_t$ using Equations (14)-(16).
  $\mathbf{y}_{t+1} = f_{\text{dyn}}(\mathbf{y}_t, \mathbf{u}_t)$
end for
while $\max\{|\delta\mathbf{y}_t|\}_{i=1}^T > \epsilon$ **do** ▷ Iterate until convergence
  for $t = T$: -1 : 1 **do** ▷ Policy update
    Compute $\widetilde{\mathbf{Q}}_t$.
    if $t == T$ **then**
      $\mathbf{S}_t = \widetilde{\mathbf{Q}}_t$
      continue
    end if
    Update $\mathbf{S}_t$ from $\mathbf{S}_{t+1}$ using Equation (22).
  end for
  for $t = 1 : T-1$ **do** ▷ Forward rollout
    Compute the optimal gain, $\mathbf{H}_t$ given $\mathbf{S}_{t+1}$.
    $\delta\mathbf{u}_t = \mathbf{H}\delta\mathbf{y}$
    $\mathbf{u}_t = \mathbf{u}_t + \delta\mathbf{u}_t$
    $\mathbf{y}_{t+1} = f_{\text{dyn}}(\mathbf{y}_t, \mathbf{u}_t)$
    $\delta\mathbf{y}_{t+1} = \mathbf{A}_t\delta\mathbf{y}_t + \mathbf{B}\delta\mathbf{u}_t$
  end for
end while

$\mathbf{H}_t$ is the optimal control gain given the control policy, $\mathbf{S}_{t+1}$:

$$\delta\mathbf{u}_t = \mathbf{H}_t\delta\widetilde{\mathbf{y}}_t. \tag{23}$$

We iterate linearization (Equation (18)) and policy update (Equation (22)) until the trajectory converges. The algorithm is summarized in Algorithm 1.

Appendix C
Analytic Form of Jacobian

We derive the analytic form of Jacobian of the dynamics and projection to formulate the iterative trajectory following. The Jacobian for the dynamics can be written as follows:

$$\mathbf{A}_t = \begin{bmatrix} \frac{\partial f_{\text{dyn}}}{\partial \mathbf{C}} & \frac{\partial f_{\text{dyn}}}{\partial P} & \frac{\partial f_{\text{dyn}}}{\partial \mathbf{q}} & \frac{\partial f_{\text{dyn}}}{\partial \mathbf{L}} \end{bmatrix}$$

$$\mathbf{B}_t = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \begin{bmatrix} \Delta t & 0 & 0 \end{bmatrix} \\ \mathbf{0}_{4\times 3} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \end{bmatrix}$$

where

$$\frac{\partial f_{\text{dyn}}}{\partial \mathbf{C}} = \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_{8\times 3} \end{bmatrix}$$

$$\frac{\partial f_{\text{dyn}}}{\partial P} = \begin{bmatrix} \mathbf{R}_z\Delta t/m \\ 1 \\ \mathbf{0}_{7\times 3} \end{bmatrix}$$

$$\frac{\partial f_{\text{dyn}}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{P\Delta t}{m}\begin{bmatrix} -2\mathbf{q}_y & 2\mathbf{q}_z & -2\mathbf{q}_w & 2\mathbf{q}_x \\ 2\mathbf{q}_x & 2\mathbf{q}_w & 2\mathbf{q}_z & 2\mathbf{q}_y \\ 0 & -4\mathbf{q}_x & -4\mathbf{q}_y & 0 \end{bmatrix} \\ \mathbf{0}_{1\times 11} \\ \mathbf{I}_4 + \begin{bmatrix} 0 & (\mathcal{J}^{-1}\mathbf{L}/2)^\mathsf{T} \\ \mathcal{J}^{-1}\mathbf{L}/2 & \left[\mathcal{J}^{-1}\mathbf{L}/2\right]_\times \end{bmatrix} \\ \mathbf{I}_3 \otimes (\mathcal{J}^{-1}\mathbf{R}\boldsymbol{\omega})^\mathsf{T}\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \end{bmatrix}$$

$$\frac{\partial f_{\text{dyn}}}{\partial \mathbf{L}} = \mathbf{I}_3$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & 0 & -4\mathbf{q}_y & -4\mathbf{q}_z \\ -2\mathbf{q}_z & 2\mathbf{q}_y & 2\mathbf{q}_x & -2\mathbf{q}_w \\ 2\mathbf{q}_y & 2\mathbf{q}_z & 2\mathbf{q}_w & 2\mathbf{q}_x \\ 2\mathbf{q}_z & 2\mathbf{q}_y & 2\mathbf{q}_x & 2\mathbf{q}_w \\ 0 & -4\mathbf{q}_x & 0 & -4\mathbf{q}_z \\ -2\mathbf{q}_x & -2\mathbf{q}_w & 2\mathbf{q}_z & 2\mathbf{q}_x \\ -2\mathbf{q}_y & 2\mathbf{q}_z & -2\mathbf{q}_w & 2\mathbf{q}_x \\ 2\mathbf{q}_x & 2\mathbf{q}_w & 2\mathbf{q}_z & 2\mathbf{q}_y \\ 0 & -4\mathbf{q}_x & -4\mathbf{q}_y & 0 \end{bmatrix}$$

where $\mathbf{q} = \begin{bmatrix} \mathbf{q}_w & \mathbf{q}_x & \mathbf{q}_y & \mathbf{q}_z \end{bmatrix}^\mathsf{T}$ and $\otimes$ is the Kronecker product.

The Jacobian of the projection can be written as follows:

$$\frac{\partial \widehat{\mathbf{x}}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{w \frac{\partial u}{\partial \mathbf{y}} - u \frac{\partial w}{\partial \mathbf{y}}}{w^2} \\ \frac{w \frac{\partial v}{\partial \mathbf{y}} - v \frac{\partial w}{\partial \mathbf{y}}}{w^2} \end{bmatrix}$$

where

$$\frac{\partial}{\partial \mathbf{y}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{C}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} & \mathbf{0}_{3\times1} & \frac{\partial}{\partial \mathbf{R}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{q}} & \mathbf{0}_{3\times11} \end{bmatrix}$$

$$\frac{\partial}{\partial \mathbf{C}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = -\mathbf{KR}$$

$$\frac{\partial}{\partial \mathbf{R}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}(\mathbf{I}_3 \otimes (\mathbf{X} - \mathbf{C})^\mathsf{T}).$$

APPENDIX D
FIRST-PERSON SPORTING ACTIVITY DATA

We collect 6 categories of sport activities from YouTube: mountain biking (MB), wingsuit flying (WF), skiing (SK), jetskiing at Lake Powell on the Coloardo River (JS), speedflying (SF), and urban bike racing in Taxco, Mexico (TX). The data and their source is summarized in Table

Inertial coefficient We approximate the inertial coefficients, e.g., mass, moment of inertia, and pivot length, based on biomechanical data [38]. Each class of activity may have different coefficient. For instance, we take into account the bike mass for mountain biking and urban biking. The inertial coefficients are summarized in Table 3.

$$\mathcal{J}_1 = \begin{bmatrix} 11.89 & -2.13 & 0 \\ -2.13 & 3.37 & 0 \\ 0 & 0 & 11.89 \end{bmatrix},$$

$$\mathcal{J}_2 = 72/80 \times \mathcal{J}_1,$$

$$\mathcal{J}_3 = \begin{bmatrix} 3.75 & 0 & 0 \\ 0 & 6.57 & 0 \\ 0 & 0 & 3.75 \end{bmatrix},$$

	MB	SK	WF	JS	SF	TX
Pivot length, l (m)	1.2	1.0	0.1	1.0	1.0	1.2
Mass, m (kg)	80	72	72	72	72	80
Moment of inertia, $\mathcal{J}$ (kg·m)	$\mathcal{J}_1$	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_1$	$\mathcal{J}_2$	$\mathcal{J}_1$

TABLE 3
Inertial coefficient

Sequence	Frame #	Time	YouTube Link
Bike 1	3297	02:00	https://www.youtube.com/watch?v=aVJ45wIUE88
Bike 2	8092	06:11	https://www.youtube.com/watch?v=khY00zN9ny0
Bike 3	8060	04:29	https://www.youtube.com/watch?v=bKUOZ4bj7Mw
Bike 4	5395	03:20	https://www.youtube.com/watch?v=WAE0rWkYULo
Bike 5	2308	01:54	https://www.youtube.com/watch?v=WmysH8uTe7U
Bike 6	2398	01:25	https://www.youtube.com/watch?v=a7drg_5RspU
Bike 7	8992	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 8	10250	08:18	https://www.youtube.com/watch?v=igp9sJkuAnU
Bike 9	8092	05:21	https://www.youtube.com/watch?v=NF-abslkZmY
Bike 10	6894	04:27	https://www.youtube.com/watch?v=7bUBNK–st4
Bike 11	3000	04:59	N/A
Bike 12	5302	03:01	https://www.youtube.com/watch?v=r3kcPqmA760
Bike 13	17983	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 14	8992	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 15	8992	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 16	8992	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 17	8992	38:52	https://www.youtube.com/watch?v=ZL0TYSfGuGM
Bike 18	6594	04:32	https://www.youtube.com/watch?v=cccuvhhQXjM
Bike 19	6594	04:32	https://www.youtube.com/watch?v=cccuvhhQXjM
Bike 20	6594	04:01	https://www.youtube.com/watch?v=ZlYAf8qFvxU
Bike 21	8100	04:58	https://www.youtube.com/watch?v=gwjXOsiVGw4
Bike 22	6594	04:27	https://www.youtube.com/watch?v=7bUBNK–st4
Bike 23	7793	05:21	https://www.youtube.com/watch?v=NF-abslkZmY
Bike 24	8992	10:42	https://www.youtube.com/watch?v=8FIeRgonxoI
Bike 25	10236	10:42	https://www.youtube.com/watch?v=8FIeRgonxoI
Bike 26	7793	06:18	https://www.youtube.com/watch?v=sIhvuZWl1s4
Bike 27	13787	08:10	https://www.youtube.com/watch?v=Wo0UuWJuT_M
Bike 28	1795	01:22	https://www.youtube.com/watch?v=FkzJ7CRS89s
Bike 29	5452	03:01	https://www.youtube.com/watch?v=r3kcPqmA760
Bike 30	47100	26:30	https://www.youtube.com/watch?v=RnUQf_kE6pw
Bike 31	14776	08:35	https://www.youtube.com/watch?v=qU5HkiTzD5I
Bike 32	14986	08:44	https://www.youtube.com/watch?v=POm-z-M_Kbg
Ski 1	1199	04:20	https://www.youtube.com/watch?v=pCcuKClUpLs
Ski 2	1799	01:29	https://www.youtube.com/watch?v=RUuVhiXe33Q
Ski 3	5395	12:07	https://www.youtube.com/watch?v=RTsQwY9PCak
Ski 4	6294	07:09	https://www.youtube.com/watch?v=8niTlZ6JnPs
Ski 5	16973	09:45	https://www.youtube.com/watch?v=pkVE1qaxX0Y
Ski 6	7000	09:45	https://www.youtube.com/watch?v=pkVE1qaxX0Y
Ski 7	5646	03:13	https://www.youtube.com/watch?v=zkR_t6p3lto
Ski 8	2905	01:40	https://www.youtube.com/watch?v=8UYwW-C6ey4
Ski 9	14349	08:16	https://www.youtube.com/watch?v=h0dhYn7cJ2o
Ski 10	3383	02:14	https://www.youtube.com/watch?v=DqH233VvxA0
Ski 11	2938	34:23	https://www.youtube.com/watch?v=SXi56fP2dQk
Ski 12	900	34:23	https://www.youtube.com/watch?v=SXi56fP2dQk
Ski 13	2098	34:23	https://www.youtube.com/watch?v=SXi56fP2dQk
Ski 14	1199	34:23	https://www.youtube.com/watch?v=SXi56fP2dQk
Ski 15	1049	34:23	https://www.youtube.com/watch?v=SXi56fP2dQk
Ski 16	4796	03:33	https://www.youtube.com/watch?v=B48B9kxcr88
Ski 17	2338	06:55	https://www.youtube.com/watch?v=Zcc77hHgXZk
Ski 18	7473	06:55	https://www.youtube.com/watch?v=Zcc77hHgXZk
Ski 19	6714	07:28	https://www.youtube.com/watch?v=Wlr3_2yRKlY
Taxco 1	4652	03:47	https://www.youtube.com/watch?v=jPYUMiOgpfw
Taxco 2	500	00:29	https://www.youtube.com/watch?v=-yrS120ttS4
Taxco 3	7770	04:27	https://www.youtube.com/watch?v=0Ytt9xUQ8Fo
Taxco 4	5500	03:50	https://www.youtube.com/watch?v=-tjplrV1O_E
Taxco 5	4600	03:25	https://www.youtube.com/watch?v=HWs5h7rmkXE
Taxco 6	6654	03:49	https://www.youtube.com/watch?v=HptS2C-1mpw
Taxco 7	6354	03:55	https://www.youtube.com/watch?v=UNDUXl1UX2w
Taxco 8	5185	03:21	https://www.youtube.com/watch?v=nZcejtAwxz4
Taxco 9	6474	04:08	https://www.youtube.com/watch?v=sITW5tr7KQg
Taxco 10	750	00:26	https://www.youtube.com/watch?v=Dgrzu94UhJ0
Taxco 11	8782	06:39	https://www.youtube.com/watch?v=li5bRjDEJ7E
Taxco 12	5065	03:31	https://www.youtube.com/watch?v=oKbUry0HGJw
Taxco 13	5310	03:38	https://www.youtube.com/watch?v=rYpJt3pMWjo
Taxco 14	6240	03:40	https://www.youtube.com/watch?v=f3okpq3fQrc
Taxco 15	3447	03:05	https://www.youtube.com/watch?v=0uvI0lHZfh8
Taxco 16	5815	03:52	https://www.youtube.com/watch?v=cwMGBC7dmd4
Taxco 17	4856	02:58	https://www.youtube.com/watch?v=yG3WHE_3k10
Taxco 18	5635	03:37	https://www.youtube.com/watch?v=vo8R6S5_zMY
Taxco 19	4826	02:46	https://www.youtube.com/watch?v=Cm0sQQWOaWg
Taxco 20	5905	03:37	https://www.youtube.com/watch?v=leHQvcU3Sko
Taxco 21	5065	03:11	https://www.youtube.com/watch?v=SPXqemrU-dc
Taxco 22	4946	03:20	https://www.youtube.com/watch?v=CkQ84Yy63wY
Taxco 23	4796	03:08	https://www.youtube.com/watch?v=JXYSQ6nSqM8

TABLE 4
Mountain biking, skiing, and urban biking sequences.

Sequence	Frame #	Time	YouTube Link
Jetski 1	2698	02:03	https://www.youtube.com/watch?v=mQg3hlilED0
Jetski 2	7372	05:01	https://www.youtube.com/watch?v=_vcXsInDlP0
Jetski 3	5068	03:22	https://www.youtube.com/watch?v=77g7Nc4wS_0
Jetski 4	9681	09:35	https://www.youtube.com/watch?v=n94gBr2sal8
Jetski 5	4646	03:42	https://www.youtube.com/watch?v=hJzaVRyv2Ys
Jetski 6	5305	13:19	https://www.youtube.com/watch?v=H7L8QcxI4WA
Jetski 7	2548	01:50	https://www.youtube.com/watch?v=e8140yfP-Vs
Jetski 8	4659	02:45	https://www.youtube.com/watch?v=zgWUZl15VF0
Jetski 9	4496	03:03	https://www.youtube.com/watch?v=iWUkuo8znoc
Jetski 10	9531	05:19	https://www.youtube.com/watch?v=CilHsYGT9d0
Jetski 11	20500	11:33	https://www.youtube.com/watch?v=xEZoacWzmn4
Jetski 12	2398	01:23	https://www.youtube.com/watch?v=J7-XzvzFtfw
Jetski 13	1499	01:48	https://www.youtube.com/watch?v=XXe5fGgyB_w
Jetski 14	5491	04:21	https://www.youtube.com/watch?v=f33kpivo-Vc
Jetski 15	17267	09:59	https://www.youtube.com/watch?v=BUdkSeC12J8
Jetski 16	2758	02:22	https://www.youtube.com/watch?v=ZHiBFYRpA04
Jetski 17	6204	04:47	https://www.youtube.com/watch?v=0aFrWsyHPK8
Jetski 18	5305	03:16	N/A
Jetski 19	5485	03:25	N/A
Jetski 20	13607	08:28	https://www.youtube.com/watch?v=3Dri6i6OvUI
Jetski 21	6984	05:25	https://www.youtube.com/watch?v=QWDFz1_p2T4
Jetski 22	11359	06:22	https://www.youtube.com/watch?v=rw0JB4Y6E60
Jetski 23	10490	06:52	https://www.youtube.com/watch?v=M_52Ier5El0
Wingsuit fly 1	1199	02:43	https://www.youtube.com/watch?v=-C_jPcUkVrM
Wingsuit fly 2	1199	02:04	https://www.youtube.com/watch?v=IM1vss7FXs8
Wingsuit fly 3	1199	11:01	https://www.youtube.com/watch?v=2fAvbqQWRWo
Wingsuit fly 4	1499	11:01	https://www.youtube.com/watch?v=2fAvbqQWRWo
Wingsuit fly 5	1619	11:01	https://www.youtube.com/watch?v=2fAvbqQWRWo
Wingsuit fly 6	2500	04:50	https://www.youtube.com/watch?v=RsV3bGPMTXo
Wingsuit fly 7	2000	04:50	https://www.youtube.com/watch?v=RsV3bGPMTXo
Wingsuit fly 8	2098	01:55	https://www.youtube.com/watch?v=bwMaqfwlER8
Wingsuit fly 9	2098	02:37	https://www.youtube.com/watch?v=_me1vneqZNE
Wingsuit fly 10	420	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 11	1440	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 12	570	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 13	900	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 14	1050	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 15	810	14:35	https://www.youtube.com/watch?v=rnvvsjstveM
Wingsuit fly 16	5695	04:10	https://www.youtube.com/watch?v=u1PUgWLRU5E
Wingsuit fly 17	4886	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 18	270	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 19	3987	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 20	720	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 21	720	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 22	2338	16:24	https://www.youtube.com/watch?v=GASFa7rkLtM
Wingsuit fly 23	960	02:37	https://www.youtube.com/watch?v=b7qwBJH9QN0
Wingsuit fly 24	2175	01:59	https://www.youtube.com/watch?v=Um4ihGIl1FE
Wingsuit fly 25	5485	03:25	https://www.youtube.com/watch?v=Zf4MztOiD4c
Wingsuit fly 26	925	01:44	https://www.youtube.com/watch?v=tNpvxVj5Nro
Wingsuit fly 27	1409	05:17	https://www.youtube.com/watch?v=-W06yAufpNQ
Wingsuit fly 28	6744	04:22	https://www.youtube.com/watch?v=vNqx8XZIWnI
Wingsuit fly 29	2500	06:21	https://www.youtube.com/watch?v=-GOLm_f36bE
Speed Flying 1	5695	03:35	https://www.youtube.com/watch?v=UwWLnaME0CI
Speed Flying 2	2398	01:31	https://www.youtube.com/watch?v=Dg55JIyBcRU
Speed Flying 3	2880	02:24	https://www.youtube.com/watch?v=226js2nxmNE
Speed Flying 4	3300	03:25	https://www.youtube.com/watch?v=oDvhIEQewhY
Speed Flying 5	7553	04:47	https://www.youtube.com/watch?v=EkFV3nNU370
Speed Flying 6	4106	02:34	https://www.youtube.com/watch?v=aOVYdwbv-5g
Speed Flying 7	570	01:30	https://www.youtube.com/watch?v=UZfdAzwZX5E
Speed Flying 8	1709	01:20	https://www.youtube.com/watch?v=RZiY2iJqIXA
Speed Flying 9	7343	04:50	https://www.youtube.com/watch?v=Gc5qEZjIMuo
Speed Flying 10	5625	04:31	https://www.youtube.com/watch?v=8bVo8vGSa5g
Speed Flying 11	3327	02:54	https://www.youtube.com/watch?v=hl3iKGd4Yqo
Speed Flying 12	2050	04:39	https://www.youtube.com/watch?v=RIyO72_E8ok
Speed Flying 13	6234	03:50	https://www.youtube.com/watch?v=_3ehKn9jsaA
Speed Flying 14	7425	04:58	https://www.youtube.com/watch?v=GKdMWEY62ZA
Speed Flying 15	6354	04:37	https://www.youtube.com/watch?v=tt0Q1BPTE7k
Speed Flying 16	3627	02:49	https://www.youtube.com/watch?v=nnfFTjiz6MM
Speed Flying 17	2368	01:31	https://www.youtube.com/watch?v=SfoaJ7b7KE8
Speed Flying 18	8302	04:57	https://www.youtube.com/watch?v=UGKmp5thh24
Speed Flying 19	4256	04:33	https://www.youtube.com/watch?v=W-VG9lEY1qY
Speed Flying 20	1199	03:13	https://www.youtube.com/watch?v=OYK4DWcFlJ8
Speed Flying 21	6264	03:43	https://www.youtube.com/watch?v=1f0rZx5IfGY
Speed Flying 22	550	06:21	https://www.youtube.com/watch?v=-GOLm_f36bE
Speed Flying 23	2925	01:37	https://www.youtube.com/watch?v=ZzVJTi8JdrY
Speed Flying 24	2278	01:32	https://www.youtube.com/watch?v=VLcPPDe-3x4
Speed Flying 25	4586	05:00	https://www.youtube.com/watch?v=WpUxLA2pw6A
Speed Flying 26	12438	07:32	https://www.youtube.com/watch?v=sazQB7G0gck
Speed Flying 27	8602	05:06	https://www.youtube.com/watch?v=857g2xz_3jc
Speed Flying 28	5150	03:34	https://www.youtube.com/watch?v=Iw0oqz2c4UA
Speed Flying 29	2098	01:26	https://www.youtube.com/watch?v=pCJpUfbhbe8
Speed Flying 30	2908	01:46	https://www.youtube.com/watch?v=aOyg_Kn1BmU

TABLE 5
Jetskiing, wingsuit flying, and speed flying sequences.