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 experiences while physically interacting with surroundings. In this paper, we focus on a problem of Force from Motion—decoding control force and torque that drive the activity—from a first person video. We use two dominant physical factors encoded in the video to understand the wearer’s activity: physical coordinate system and active force component. (1) A physical coordinate system aligned with gravity in metric scale is a prerequisite for force computation. We learn the gravity cue that exists in a natural image to predict the 3D gravity direction given a sequence of images. The sense of physical scale is revealed to us when the body is in a dynamically balanced state. We compute the unknown physical scale in metric unit by leveraging the torque equilibrium at a banked turn. (2) The active force and torque act on the wearer’s body, reflected in the 3D camera egomotion. We model the dynamics using an inverted pendulum that is simple yet expressive enough to describe force interactions. The optimal active control components that generate the camera motion are computed such that it minimizes 2D reprojection error. As a by-product, we can also recover the passive forces such as air drag, friction, and ground reaction/lifting force. 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 underlying control of actions. Most computer vision systems have focused on direct observation from a camera looking at us from a third person perspective such as surveillance cameras. This camera produces a view that has a limited visual access to muscle movements due to heavy occlusion when the camera is close, or missing details when the camera is far. Furthermore, for many activities, such as wingsuit flying in Figure 1(a), placing a third person camera closer enough to record the muscle movement is impossible.

We tackle human activity understanding from a different perspective: using a first person video mounted on an actor’s head or body without directly observing any of the body parts. The key insight is that the camera experiences all the forces acting on the actor’s body. These forces include passive force (e.g., gravity and air resistance) and active force that the actor exerts through her/his muscle movements.

In this paper, we study a problem of Force from Motion—decoding ‘how’ active components of physical force and torque control the movement of the camera wearer—from a first person video. We analyze the visual semantics of the video to recognize the physical coordinate system in the gravitational field, and leverage the 3D camera egomotion to inversely compute control forces exerted by the wearer. For instance, in a wingsuit flying first person video1, 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).

Extracting the controls of activity involves with three fundamental challenges inherent in first person videos. a) Complexity of physical interactions: many factors such as joint angle configurations affect the dynamics of the camera wearer, which is not visible; b) Geometric ambiguity: structure from motion reconstructs camera motion up to scale and orientation which is not a physically meaningful coordinate (e.g., gravitational field and metric scale) that Newtonian dynamics applies; c) State observability: the video only observes net camera egomotion as a resultant of the active and passive force interactions.

We address these challenges by modeling the dynamics of a first person video using an inverted pendulum; this model is simple yet expressive enough to relate physical forces with visual egomotion. We focus on decoding three dominant physical quantities: gravity, momentum, and force/torque.

First, motion is driven by gravity which can be estimated from two visual cues in a first person video. (1) Natural images encode gravity because it affects how physical environment is formed, i.e. trees and buildings are usually vertical, water surface normal aligns with gravity direction, and horizon is perpendicular to it. We learn gravity visual semantics embedded in 2D image using a convolutional neural network designed to recognize the orientation of the image. (2) Gravity is reflected in camera rotational egomotion. For instance, the camera angle forms a bank angle when turning with respect to the gravity. We learn the 3D gravity given a sequence of 3D camera orientations using a long short-term memory network. We integrate these two cues to predict a global 3D gravitational field using a maximum a posteriori estimate.

Second, how fast we are going (speed or momentum) in a physics unit (m/s or kg·m/s) allows us to relate with how much force is applied via Newtonian dynamics. The absolute scale of our motion is revealed to us when the body is in a dynamically

1. https://www.youtube.com/watch?v=IM1vss7FXs8
balanced state. During a banked turn, the torques produced by centripetal force and gravity force are balanced with the body leaning angle to satisfy moment equilibrium. This physical constraint together with the known gravitational acceleration constant, i.e., 9.81 m/s², allows us to compute the physical scale exactly.

Third, we decompose the force and torque into semantically meaningful components, i.e., active components exerted by the camera wearer such as twisting body orientation in flying and pedaling in biking and passive components exerted by the environment such as air drag. For each type of first person sport video, we construct a rigid body dynamics using an inverted pendulum model and compute the active components by solving for the optimal control, i.e., the geometric reprojection error is minimized with respect to egomotion governed by the dynamics. As a by-product, we can also recover the passive forces such as air drag, friction, and ground reaction/lifting force. This inverted pendulum model is highly predictive and generalizable. It can be applied to different activities such as mountain biking, skiing, and speed flying with a few minor modification (coefficient change).

In total, our system takes an input, a first person sport video, and outputs active force and torque profile in a physical coordinate system aligned with the gravitational field as shown in Figure 1(c). We predict the 3D gravity direction by integrating 2D prediction via a convolutional neural network and recover physical scale using the roll moment equilibrium. These factors are embedded in the optimal control with reprojection error minimization, which produces a plausible control forces that generate visual dynamics of the first person video.

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 [53] but also computational sport analytics [48], 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. Despite 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 [40] 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; (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 mountain biking, urban bike racing, skiing, speedflying with parachute, and wingsuit flying where inertial measurements are not accessible.

### 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 [25] 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. [54] maximized a posterior 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 noted by Taylor [57]. 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 [46]. A factorization based approach [7], [60] was used by Yan and Pollefeys [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. [42] used trajectory bases.

Metaxas and Terzopoulos [35] 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 [26], [45], [47], and motivated many vision tasks such as understanding fixation point [32], identifying eye contact [70], and localizing joint attention [16], [41].

A first person camera egomotion 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. [27] to categorize and segment a first person sport video in a unsupervised manner. Coarse-to-fine motion models [51] and a pretrained convolutional neural network [52] 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. [28] stabilized first person footage via 3D reconstruction of camera egomotion. In a social setting, joint attention was estimated via triangulation of multiple camera optical rays [41], [43] and the estimated joint attention was used to edit social video footage [3].

Another information that the first person camera captures is egomotion or scene motion. Pirsivash and Ramanan [44] used an object centric representation and temporal correlation to recognize active/passive objects from an egocentric video, and Rogne et al. [49] leveraged a prior distribution of body and hand coordination to estimate poses from a chest mounted RGBD camera. Lee et al. [30] 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. [45] 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 [39], [55] and objects to interact [6]. Gaze direction can be precisely estimated from visual semantics and 3D motion of first person videos [32], important objects can be detected [30], [44], visual transformation can be predicted through egomotion [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. We leverage two motion cues: 1) 3D reconstruction from egomotion, and 2) gravity and scale recovery from exomotion. 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

The input of our algorithm is a first person video of sporting activities heavily driven by camera egomotion such as mountain biking, skiing, jetskiing, speedflying, and wingsuit flying. The output is control active force and torque in a physically meaningful coordinate system (aligned with gravity and scaled in metric units). The pipeline of the algorithm is summarized in Figure 2. We reconstruct the 3D camera trajectory using structure from motion, which has inherent scale and orientation ambiguity (Section 4.2). The reconstruction is transformed to a physical coordinate system by estimating gravity and scale. We predict the 3D gravity using visual semantics of a video by incorporating with 3D camera trajectory in Section 4.3. For scale, we recover the metric scale factor by leveraging the moment balance at a bank turn in Section 4.4. Given the physical coordinate system, we compute the optimal control that drives the activity using an inverted pendulum model in Section 5.
Fig. 2. Our system takes a first person video of sporting activities and estimate control active force and torque that drive the 3D camera egomotion. We recognize a physical coordinate system by estimating gravity and metric scale. Based on the coordinate, we compute the optimal force acting on the wearer’s body by minimizing reprojection error.

4 Physical Coordinate System

A physical coordinate system is a prerequisite to model force interactions around the camera wearer. In this section, we recognize a physically meaningful coordinate system (aligned with gravity and scaled in metric units), which cannot be deduced by standard structure from motion [19] due to its inherent scale and orientation ambiguity.

4.1 3D Camera Trajectory Reconstruction

We represent an each image using a triplet, \( \{t_i, R_i, C_i\}_{t=1}^T \) where \( t_i \in \mathbb{R}^{h \times w \times 3} \), \( R_i \in \text{SO}(3) \), and \( C_i \in \mathbb{R}^3 \) are the RGB image with width \( w \) and height \( h \), 3D orientation, and optical center of the camera at the \( t^\text{th} \) time instant. We recover the 3D camera pose \( \{R_t, C_t\} \) and point cloud, \( \{X_p\}_{p=1}^P \) where \( P \) is the number of 3D points using structure from motion [19]. To handle a large scale reconstruction (1 minute video produces 1800 images), we apply bundle adjustment in a sliding window fashion. For instance, in the second column of Figure 2, we reconstruct \( T = 3,658 \) images (~2 minutes) with \( P = 1,298,317 \) points.

The \( p^\text{th} \) 3D point, \( X_p \in \mathbb{R}^3 \) is projected onto the camera projection matrix, \( P_t = KR_t [I_3 - C_t] \in \mathbb{R}^{3 \times 4} \) to form the 2D projection, \( \hat{x}_{t,p} \in \mathbb{R}^2 \) where \( K \) is the global intrinsic parameter of the camera encoding focal length and principal points, i.e., \( \hat{x}_{t,p} = \left[ P_t^1 \hat{X}_p / P_t^2 \hat{X}_p / P_t^3 \hat{X}_p \right] \) where \( \hat{X} \) is the homogeneous representation of \( X \) and \( P_t^i \) indicates the \( i^\text{th} \) row of \( P_t \).

Bundle adjustment minimizes the reprojection error between the projection, \( \hat{x} \), and 2D feature location, \( x \):

\[
\text{minimize} \{R_t, C_t\}, \{X_p\} \sum_{t=1}^T \sum_{p=1}^P \delta_{t,p} \|x_{t,p} - \hat{x}_{t,p}\|^2, \tag{1}
\]

where \( \delta_{t,p} \) is the Kronecker delta function that produces 1 if \( X_p \) is visible from the \( t^\text{th} \) image, and 0 otherwise. Equation (1) is an unconstrained nonlinear least squares optimization and can be efficiently solved by exploiting the sparse nature of the original equation [34].

We will re-parametrize Equation (1) in terms of force and torque represented in physical metric unit, e.g., distance in m, force in N, torque in Nm, and angle with respect to the gravitational field. However, this is challenging because the 3D camera trajectory is reconstructed up to scale and rotation [19]: the 3D camera pose of the first image is often set to the origin (\( C_1 = 0 \) and \( R_1 = I_3 \)), and the distance between the first and second images is set to 2. We apply the global intrinsic parameters as 95% of first person videos are recorded by GoPro camera at the same resolution (1280×720). For lens distortion, we use fish-eye parametric model [13].

Fig. 3. We rectify a first person image based on 3D camera egomotion such that the projection of the future camera location (blue point) maps consistently to the center of the image. This image rectification allows learning visual semantics of gravity efficient.

The unit distance (\( \|C_1 - C_2\| = 1 \)). This scale and orientation ambiguity is inherent in structure from motion, which prevents from recovering physically meaningful force and torque acting on the wearer’s body. In the subsequent sections, we will present a method to calibrate orientation (gravitational field) and physical scale (metric units) given a first person video.

4.2 First Person Image Rectification

The visual pattern of a first person video is highly dependent on the camera placement (location and orientation). For an extreme example, a video could be oriented by landscape or portrait mode (flipping xy axis), which makes learning visual semantics challenging. In this section, we rectify a first person image with respect to the wearer’s body motion (the camera trajectory).

A key insight is that the visual scene motion appears consistently with respect to its camera egomotion regardless of the placement. We rectify a first person image using the instantaneous velocity \( v_t \) and (centripetal) acceleration \( a_t \):

\[
v_t = \frac{C_t - C_{t-1}}{\Delta t}, \quad a_t = \frac{C_{t+1} + C_{t-1} - 2C_t}{\Delta t^2}. \tag{2}
\]

We define a global rotational transformation, \( M \in \text{SO}(3) \), per sequence such that:

\[
M^* = \arg\min_{M \in \text{SO}(3)} \frac{\theta_M}{2 \sin \theta_M} \left\| M - M^T \right\|_F \tag{3}
\]

subject to \( M^2 v_t / v_t = 0 \), \( M^2 a_t / a_t = 0 \), \( \forall t \),

where \( \theta_M = \cos^{-1}\left(\frac{\text{trace}(M)-1}{2}\right) \) is the angle in axis-angle representation of rotation matrix \( M \), and \( M^2 \) is the second row of \( M \). The constraint enforces that the resulting y axis of the first person camera is aligned with the surface normal of the
Fig. 4. 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 function using a convolutional neural network.

plane spanned by instantaneous velocity and acceleration, i.e., \( \frac{v}{\|v\|} \times \frac{a}{\|a\|} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \). Equation (3) finds the rotation that minimizes geodesic distance in the SO(3) manifold constrained by the camera translational motion. We solve Equation (3) using a gradient decent optimization initialized by the geometric mean of \( \mathbf{M}^2 = \frac{v}{\|v\|} \times \frac{a}{\|a\|} \) and \( \mathbf{M}^3 = \frac{v}{\|v\|} \). Then, \( \mathbf{M} = \mathbf{M}^2 \times \mathbf{M}^3 \).

Given \( \mathbf{M} \), we transform the structure from motion coordinate, i.e., \( \mathbf{X}_M = \mathbf{M} \mathbf{X}, \mathbf{R}_M = \mathbf{R} \mathbf{M}^T, \) and \( \mathbf{C}_M = \mathbf{M} \mathbf{C} \). This first person coordinate based rectification corrects the rotational bias caused by the choice of the initial pair of images. We also rectify the image via a homography, \( \mathbf{H} = \mathbf{KMK}^{-1} \), i.e., \( \mathbf{I}_M = \mathbf{Hx} \). We abuse notations by dropping subscript \( M \) for subsequent sections. Figure 3 shows the rectification of a first person image (left) based on the camera egomotion where the blue point illustrates the projection of the future camera location aligned with velocity, \( v \). Note that the projection forms always at the center of the image, which produces a consistent spatial arrangement around the camera wearer regardless of the camera placement. This rectification allows learning visual semantics of gravity efficient (Section 4.3).

4.3 Gravitational Field Estimation

The gravitational field is a dominant physical quantity that drives motion, i.e., the potential energy is converted to kinetic energy. For instance, a downhill mountain biker/skier uses the gravity to accelerate their motion to jump higher. In particular, the dynamics of first person sporting activities such as mountain biking, wingsuit flying, and skiing is heavily influenced by the gravitational field. In this section, we estimate the gravitational field from a sequence of first person images.

While structure from motion is completely ignorant of global orientation, 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], [38]. In this section, we leverage such image cues learned by a convolutional neural network to predict 3D gravitational field from a series of images.

We represent the gravitational field using a global unit vector in 3D, \( \mathbf{G}(\phi_1, \phi_2) = [\sin \phi_1 \cos \phi_2 \sin \phi_1 \sin \phi_2 \cos \phi_1]^T \in \mathbb{S}^2 \) where a point in a unit sphere parametrized by polar \( \phi_1 \) and azimuthal \( \phi_2 \) angles in a spherical coordinate. This coordinate is seen from the first person coordinate that applies for all images in a video.

We formulate the 3D gravity prediction problem as the maximum a posteriori (MAP) estimation using visual and motion cues, \( \{\mathcal{I}_t, \mathbf{R}_t\}_{t=1}^T \):\[
\mathbf{G}^* = \arg \max_{\mathbf{G} \in \mathbb{S}^2} p(\mathbf{G}|\mathcal{I}_1, \cdots, \mathcal{I}_T, \mathbf{R}_1, \cdots, \mathbf{R}_T) \tag{4}
\]
where \( p(\mathbf{G}|\mathbf{R}_1 \cdots \mathbf{R}_T) \) is a gravity prior provided by camera motion, and \( p(\mathcal{I}_t|\mathbf{G}, \mathbf{R}_t) \) is a likelihood of the visual semantics, given the 3D gravity, i.e., how well the 3D gravity direction is aligned with the image. These motion and visual cues provide two independent information: 1) the motion cue is provided by a 3D kinematic relationship of the gravity with respect to the camera motion, e.g., the dominant forward movement is driven by gravity and the banked angle forms when centripetal force applied (turning); 2) the visual semantic cue is extracted from the image RGB data, e.g., trees and horizon. Equation (4) integrates these two cues over the entire video to predict the global 3D gravity parameters, \( \phi_1 \) and \( \phi_2 \).

Predictions on multiple images are consolidated by the 3D reconstructed camera orientations. Figure 4 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 2D projection. Each image produces a streak in a likelihood distribution (middle)—any gravity direction along the streak is projected onto the same direction in 2D, i.e., if ones goes straight without changing camera orientation, the streak remains constant. This ambiguity can be further resolved by taking into account more images that moves different heading directions. The product of multiple image predictions in Equation (4) by leveraging the 3D reconstructed camera orientations can collapse the streak into a unimodal distribution (right).

4.3.1 Learning Gravity Prior from Camera Motion

The camera motion is highly affected by the gravitational field, e.g., the body is often oriented upright, the dominant forward movement is driven by gravity, and the banked angle forms when centripetal force applied (turning). 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:

\[
\tilde{\mathbf{G}}_t = g_{\text{mot}}(q_{t+1}^{+\Delta t}, \cdots, q_t^{+\Delta t}; \mathbf{w}_{\text{mot}}), \tag{5}
\]
where \( q_{t+\Delta t}^{+\Delta t} \) is the relative rotation from the \( t \)th image to \( (t + \Delta t) \)th image, \( \mathbf{R}_{t+\Delta t}^T \mathbf{R}_t^T \) in quaternion representation. \( g_{\text{mot}} \) encodes the dynamics of first person rotation over time to predict the gravity direction, \( \tilde{\mathbf{G}}_t \) parametrized by \( \mathbf{w}_{\text{mot}} \). We learn \( \mathbf{w}_{\text{mot}} \) using a long short-term memory (LSTM) [20] and define the loss using the angle distance in \( \mathbb{S}^2 \):

\[
L_{\text{mot}}(\tilde{\mathbf{G}}, \mathbf{G}_{gt}) = 1 - \tilde{\mathbf{G}}^T \mathbf{G}_{gt},
\]
where \( \mathbf{G}_{gt} \) is the ground truth gravity direction from training data in Appendix A.
Given a sequence of predictions \( \{ \hat{G}_t \}_{t=1}^{T} \), we model a prior distribution of the 3D gravity direction in Equation (4) using a mixture of von Mises-Fisher distributions:

\[
p_{\text{mot}}(G|R_1, \cdots, R_T) \
\approx \sum_{m=1}^{M} \frac{\kappa_m}{4\pi \sinh \kappa_m} \exp \left( \kappa_m G^T g_m \right)
\]

where \( \{g_m, \kappa_m\} \) is a set of modes and concentration parameters, and \( M \) is the number of modes. We learn \( \{g_m, \kappa_m\} \) from \( \{G_t\}_{t=1}^{T-\Delta t} \) using an Expectation-Maximization algorithm [14].

### 4.3.2 Learning Gravity Likelihood from Visual Semantics

There exist visual gravity cues in a natural image, and we learn such visual semantics from first person images. The image likelihood measures how well the projected 3D gravity direction onto the \( t \)th image agrees with the first person visual semantics, \( I_t \):

\[
p_{\text{vis}}(I_t|G, R_t) = \mathcal{L}_{\text{vis}} \left( I_t \bigg| \theta_g = \tan^{-1}\left( \frac{R_{1,1}^T G}{R_{1,2}^T G}; w_{\text{vis}} \right) \right) .
\]

where \( \mathcal{L}_{\text{vis}} \) is a likelihood distribution of the image over its orientation \( \theta_g \) with respect to the gravity parametrized by \( w_{\text{vis}} \), i.e., \( \theta_g = 0 \) if the gravity is aligned with the Y axis of the image. \( R^1 \) and \( R^2 \) are the first and second rows of the rotation matrix, \( R \). We model the likelihood distribution, \( \mathcal{L}_{\text{vis}}(I_t|\theta_g; w_{\text{vis}}) \) using a convolutional neural network (CNN). We cast this learning gravity semantics as a image classification problem where the class corresponds to the image orientation, i.e., we discretize angle with 1 degree resolution across \( [-\pi/2, \pi/2] \). We use the softmax loss and the FC8 layer of AlexNet [29] to compute the likelihood distribution. The existing model trained on object classification in ImageNet [50] is used as an initialization of the network where a resized image (320×180) is used as an input of the network.

Predictions on multiple images are consolidated by the product of the likelihood, \( \prod_{t=1}^{T} p_{\text{vis}}(I_t|G, R_t) \) in Equation (4). Note that a single image cannot predict the 3D gravity direction due to 2D projection. Each image produces a streak in a likelihood distribution as shown in Likelihood of Figure 4—any gravity direction along the streak is projected onto the same direction in 2D. Combining multiple image predictions by leveraging the 3D reconstructed camera orientations can collapse the streak into a unimodal distribution. 3.

Figure 5 illustrates the likelihood of the gravity direction learned by CNN as shown in the red heatmap and dark red triangle shows prediction \( \theta_g \). We also encode the per pixel evidence of the gravity prediction using a fully convolutional neural network [33] 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 bicyclist.

### 4.4 Physical Scale Recovery

Due to scale ambiguity of structure from motion, the quantities are not represented in metric units, e.g., speed in m/s, force in N, and torque in Nm. There exists an unknown scale factor, \( \alpha \), i.e. \( v = \alpha v_{\text{sfm}} \) and \( a = \alpha a_{\text{sfm}} \) where \( v \) and \( a \) are the linear velocity and acceleration in physical metric (m/s and m/s²) while \( v_{\text{sfm}} \) and \( a_{\text{sfm}} \) are with an ambiguous scale from structure from motion in Equation (2). To recover \( \alpha \), a reference physical quantity such as height of the objects [11], the baseline of stereo cameras, or external IMU sensor [36] needs to be known. In this section, we exploit the constant gravitational acceleration, \( g = 9.81 \text{m/s}^2 \), to compute \( \alpha \).

At a banked turn, the gravitational force balances with the centripetal force in relation to the banked angle, \( \theta_b \), i.e., the moment balance equilibrium at the pivot point as shown in Figure 7:

\[
L m \sin \theta_b - L F_C \cos \theta_b = 0,
\]

where \( L \) is the length from the pivot point to the center of mass, \( m \) is mass, \( F_C \) is the lateral force induced by centripetal acceleration, \( F_C = \alpha m \), \( a_{\text{sfm}} \), and \( \theta_b \) is the banked angle. The fact that \( g \) is constant allows solving for \( \alpha \):

\[
\alpha = \frac{g}{|a_{\text{sfm}}|} \tan \theta_b,
\]

where the banked angle \( \theta_b \) can be computed via the 3D gravity, i.e., \( \theta_b = \cos^{-1} \frac{G^T R^2}{|R^2|} \). This moment balance equation is not limited to the dynamics on the ground (biking and skiing) but the aerodynamics (wingsuit flying) as it needs to balance between weight and lifting force to produce centripetal acceleration. As shown in Figure 6(a), it is possible to measure \( \alpha \) whenever the first person camera experiences centripetal acceleration. The red points illustrate the ratio between the banked angle, \( \tan \theta_b \), and centripetal acceleration, \( |a_{\text{sfm}}| \). The distribution of the points form a line where the slope is the scale factor, \( 1/\alpha \). Note that \( \tan \theta_b < 0 \) and \( \tan \theta_b > 0 \) indicate right-turn and left-turn, respectively. We find the slope of the line that best represents the point distribution by finding the median \( \alpha \). Figure 6(b) shows the torques produced by the scale factor, and two torques are roughly canceled out due to correct \( \alpha \), i.e., \( T_R + T_N = 0 \). This allows us to reconstruct the terrain elevation and speed in metric units as shown.
Fig. 6. (a) We plot the scale factor, a ratio between centripetal acceleration computed by structure from motion and banked angle, $a = \frac{a_k}{\tan \theta}$. The slope of red point distribution is a scale factor where we find the median of the distribution. (b) We verify the moment balance at a banked turn, $T_N + T_R = 0$. (c) The recovered gravity direction and scale allow us to compute the terrain elevation and speed in metric units.

in Figure 6(c). Note that the speed profile is physically meaningful, i.e., average speed of the mountain biking ranges between 1-6 m/s.

## 5 Force from Motion

Net force and torque act on the center of mass of the camera wearer, which affects linear and angular acceleration. The physical coordinate system in Section 4 allows relating 3D reconstruction with the force via Newtonian dynamics. In this section, we formulate the dynamics to identify the active components of force and torque that drive the first person activity.

We recover the series of active force and torque (control input) $\{u_i\}_{i=0}^{T-1}$ by reformulating Equation (1) that minimizes reprojective error:

$$
\min_{\{u_i\}_{i=0}^{T-1}} \sum_{t=1}^{T} \sum_{p=1}^{P} \delta_{t,p} ||x_{t,p} - \bar{x}_{t,p}||^2 + L_{\text{reg}}(u_1, \ldots, u_{T-1})
$$

subject to $y_{t+1} = f_{\text{dyn}}(y_t, u_t)$,  

where the camera projection matrix is parametrized by the state $\{y_t\}_{t=0}^{T}$, i.e., $P(y_t) = KR(y_t) \begin{bmatrix} I_3 & -C(y_t) \end{bmatrix}$. $f_{\text{dyn}}$ is discrete Newtonian dynamics that describes the relationship between state and force, i.e., the active force and torque generate a camera trajectory that satisfies 2D image projection at discrete time instances (reprojection error). $L_{\text{reg}}$ is a regularization of control, which is needed to find the most plausible active force and torque.

### 5.1 Equation of First Person Motion

We model the dynamics of a first person camera, $f_{\text{dyn}}$ in Equation (9) using a 3D inverted pendulum model as shown in Figure 7. We represent force and torque in terms of the first person body coordinate, $\{B\}$. We define the state and control:

$$
y = \begin{bmatrix} C^T & P & q^T & L^T \end{bmatrix}^T \in \mathbb{R}^{11}, \quad u = \begin{bmatrix} F_T & T_Y & T_R \end{bmatrix}^T \in \mathbb{R}^{3},
$$

where $F_T$ is the thrust force applied along the velocity direction, $v$, ($Z$ axis of the first person coordinate) using pedaling and braking, $T_Y$ is the yaw torque applied through steering wheel, and $T_R$ is the roll torque to balance the posture, which is the residual of bank turn balance in Equation (7). $P$ is the linear momentum along the instantaneous velocity, $P = m||v||$, and $L$ is the angular momentum, $L = R^1 \omega$. $J$ is moment of inertia and $\omega$ is angular velocity in the first person coordinate.

We model the discrete dynamics as follow:

$$
f_{\text{dyn}}(y, u) = \begin{bmatrix} C + R^3 P \Delta t/m \\
   P + (F_R + F_T) \Delta t \\
   q + (J^{-1} L) q/2 \\
   L + \begin{bmatrix} T_Y \\
   T_R \end{bmatrix} \Delta t
\end{bmatrix},
$$

where

$$
F_R = -R_R - D + mg G^T R^3,
\quad \tau_R = -mg L G^T R^1 - C G^T R^2
$$

4. $L_{\text{reg}}(u_1, \ldots, u_{T-1}) = \int_1^{T-1} u \, dt$ is often used for a linear quadratic regulator.
5. We approximate the center of mass with the 3D camera pose, which is empirically validated in Appendix B.
6. 3D point cloud $X$ can be a part of the state [58] while we set it constant computed by structure from motion.
and \( \hat{R}_{\text{R}} = \mu_R mg \) is the rolling/sliding friction along the Z axis of the first person coordinate, and \( \mu_R \) is rolling friction coefficient. \( D = 0.5C_D \rho A |\dot{v}|^2 \) is the air drag force where \( C_D \approx 1.0, \rho = 1.23 \text{ kg/m}^3 \), and \( A \) are air drag coefficient, air density, and cross sectional area perpendicular to the velocity, respectively. \( F_C \) is centripetal force in Equation (7). \( R_t \) is the \( t \)th row of \( R \).

Due to action-reaction, the passive force and torque can be computed as:

\[
F_N = mg \sin \theta_p = mgG^TR^2 \\
R_S = \mu_S F_N \cos \theta_p \\
T_p = -L_1^1 / \Delta t
\]

where \( \theta_p = \cos^{-1} (G^TR^3) \) and \( L_1^1 \) is the first element (pitch direction) of the angular momentum. \( F_N, R_S, \) and \( T_p \) are normal or lifting force, sliding friction with \( \mu_S \) friction coefficient, and passive torque along the pitch direction created by an unbalance impact between two wheels in a bicycle as shown in Figure 7. Note that the biking activity is used for an illustrative purpose while this dynamics can generalize for various sporting activities such as skiing, jet-skiing, 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 C for the activity dependent parameters.

5.2 Camera Trajectory Following via Linearization

Equation (9) is highly nonlinear and no closed form solution exists. A proxy of the problem in the optimal control theory [56] is a nonlinear trajectory following solved via iterative linear quadratic regulator (iLQR) [31], [59]. Note that in our camera trajectory following problem, nonlinearity exists in not only the dynamics but also the cost function (projection). We present a new iterative algorithm for the camera trajectory following designed to minimize reprojection error, which eventually converges to Equation (9). We iterate two processes: (1) compute the control policy by linearizing dynamics near a nominal trajectory, \( \{y_t\}_{t=1}^T \); (2) update the nominal trajectory based on the locally optimal control input, \( \{u_t\}_{t=1}^{T-1} \).

**Linearizing dynamics** Let \( \delta y_t \) and \( \delta u_t \) be the deviation from the nominal state and control input, respectively, i.e., \( y_{t+1}^i = y_t^i + \delta y_t \) and \( u_{t+1}^i = u_t^i + \delta u_t \) where the superscript represents the iteration number. We linearize the state dynamics in Equation (10) using the first order Taylor expansion:

\[
y_{t+1}^i - y_t^i = \delta y_{t+1} = A_t \delta y_t + B_t \delta u_t,
\]

(11)

where the analytic form of \( A_t = \frac{\partial f_t}{\partial y_t} \) \( y_t \in \mathbb{R}^{11 \times 11} \) and \( B_t = \frac{\partial f_t}{\partial u_t} \) \( u_t \in \mathbb{R}^{1 \times 3} \) can be found in the Appendix D.

**Linearizing reprojection error** We approximate the 2D projection, \( \hat{x} \), near nominal state trajectory as follow:

\[
\hat{x}(y + \delta y) \approx \hat{x}(y) + \frac{\partial \hat{x}}{\partial y} \delta y,
\]

(12)

where \( \frac{\partial \hat{x}}{\partial y} \) is the Jacobian of the projection where its analytic form can be found in the Appendix D.

This allows linearizing the reprojection error in Equation (9):

\[
\sum_{t=1}^T \sum_{p=1}^p \delta y_{t,p}\|x_{t,p} - \hat{x}_{t,p}\|^2 \approx \sum_{t=1}^T \|e_t - C_t \delta y_t\|^2,
\]

(13)

where

\[
e = \begin{bmatrix}
\delta y_{t,1}(x_{t,1} - \hat{x}_{t,1}) \\
\vdots \\
\delta y_{t,p}(x_{t,p} - \hat{x}_{t,p})
\end{bmatrix},
C_t = \begin{bmatrix}
\delta y_{t,1} \frac{\partial \hat{x}_{t,1}}{\partial y} \\
\vdots \\
\delta y_{t,p} \frac{\partial \hat{x}_{t,p}}{\partial y}
\end{bmatrix}.
\]

**Recursive cost function** By combining Equation (9) and (13), we can derive a recursive cost-to-go function:

\[
J_t = \delta \bar{y}_t^T Q_t \delta \bar{y}_t + \delta \bar{u}_t^T W \delta \bar{u}_t + \min_{\delta u_t} J_{t+1},
\]

(14)

where \( J_t \) is accumulated cost at the \( t \)th time instance and \( \bar{y} \) and \( \bar{u} \) are the homogeneous representation of \( y \) and \( u \), respectively. The first two terms in Equation (14) measures the reprojection error and regularization loss where

\[
Q_t = \begin{bmatrix}
C_t^T C_t & -C_t^T e_t \\
e_t^T C_t & e_t^T e
\end{bmatrix},
W_t = \begin{bmatrix}
I_t & -d_t \\
-d_t^T & d_t^T d_t
\end{bmatrix},
\]

where \( d_t = u_t^i - u_{t-1}^i \). The regularization via \( W_t \) prevents \( u_t^i \) deviates too much from the nominal control input \( u_{t-1}^i \).

The accumulated cost, \( J_t \) in Equation (14) can be rewritten in a closed form based on the current nominal state, \( y_t \), i.e., \( J_t = \delta \bar{y}_t^T S_t \delta \bar{y}_t \) where

\[
S_t = Q_t + H_t^T W H_t + \left( \bar{A}_t + \bar{B}_t H_t \right)^T S_{t+1} \left( \bar{A}_t + \bar{B}_t H_t \right)
\]

\[
H_t = - \left( W + \bar{B}_t^T S_{t+1} \bar{B}_t \right)^{-1} \bar{B}_t^T S_{t+1} \delta \bar{A}_t,
\]

(15)

where \( \bar{A}_t, \bar{B}_t, \delta \bar{A}_t \), and \( \delta \bar{B}_t \) represent the transition matrix for the homogeneous representation, \( \bar{y}_t \), i.e., \( \delta \bar{y}_{t} = \bar{A}_t \delta \bar{y}_{t-1} + \bar{B}_t \delta \bar{u}_t. \) \( H_t \) is the optimal control gain given the control policy, \( S_{t+1} \):

\[
\delta \bar{u}_t = H_t \delta \bar{y}_t.
\]

(16)

We iterate linearization (Equation (11)) and policy update (Equation (15)) until the trajectory converges. The algorithm is summarized in Algorithm 1.

5.3 Control Input Initialization

Our iterative approach to recover the series of the control input finds a local optimal solution that requires a valid initialization. We use the reconstructed camera trajectory to recover the full state, \( \{y_t\}_{t=1}^T \), and then, compute the initial estimate of control input, \( \{u_t\}_{t=1}^T \) by solving Equation (10) for the linear and angular momentum, \( P_t \) and \( L_t \):

\[
F_T = \frac{P_{t+1} - P_t}{\Delta t} - F_R
\]

(17)

\[
T_Y = \frac{L_{Y}^{t+1} - L_Y^t}{\Delta t}
\]

(18)

\[
T_R = \frac{L_{R}^{t+1} - L_R^t}{\Delta t} - \tau_R
\]

(19)

where \( L_Y^t \) and \( L_R^t \) are the yaw and roll elements of the angular momentum. Note that Equations (17)-(19) do not take into account the dynamics, input control sequence, and reprojection error. This initialization of control input is used to regenerate the nominal states, and further refined through the iterative policy update.
Algorithm 1 Camera trajectory following

Input: $X_j \leftrightarrow x_{1:3}$ and $\{C_t, R_t\}_{t=1}^{T}$

1. $y_1 = \begin{bmatrix} C_1 \mid P_1 \mid q_1 \mid L_1 \end{bmatrix}^T$
2. $\delta y_1 = 0^T$
3. for $t = 1 : T - 1$
5. $y_{t+1} = f_{dyn}(y_t, u_t)$
6. end for
7. while $\max \{|\delta y_t|\} > \epsilon$ do
8. for $t = T : -1 : 1$
9. Compute $Q_t$.
10. if $t == T$ then
11. $S_t = Q_t$
12. continue
13. end if
14. Update $S_t$ from $S_{t+1}$ using Equation (15).
15. end for
16. for $t = 1 : T - 1$
17. Compute the optimal gain, $H_t$ given $S_{t+1}$.
18. $\delta u_t = H_t \delta y_t$
19. $u_t = u_t + \delta u_t$
20. $y_{t+1} = f_{dyn}(y_t, u_t)$
21. $\delta y_{t+1} = A_t \delta y_t + B \delta u_t$
22. end for
23. end while

6 RESULT

We evaluate our algorithm in terms of three key physical quantities: gravity, scale, and force/torque.

6.1 Validation via Synthetic Data

We validate our algorithm in Equation (9) using synthetic data. We simulate the first person motion sliding down in a downhill slope as shown in Figure 8. We generate random 1,200 3D points, $X$, and 10 seconds of motion (300 frames). At least 40 random points per image in front of the camera are chosen to be visible, $\delta x_{1:3}$, and the ground truth control input, $\{u_t\}_{t=1}^{T-1}$, is designed to produce a natural S curve along the downhill. In Figure 8, the color on the trajectory represents the speed, and yellow lines indicate the direction and magnitude of its curvature.

We validate our algorithm in Figure 9 using synthetic data. We simulate the first person motion sliding down in a downhill slope as shown in Figure 8. We generate random 1,200 3D points, $X$, and 10 seconds of motion (300 frames). At least 40 random points per image in front of the camera are chosen to be visible, $\delta x_{1:3}$, and the ground truth control input, $\{u_t\}_{t=1}^{T-1}$, is designed to produce a natural S curve along the downhill. In Figure 8, 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, $\theta_p$: $y = \begin{bmatrix} X \mid Y \mid P \mid \theta \mid J \omega \end{bmatrix}^T$ and $u = \begin{bmatrix} F_T \mid T_Y \end{bmatrix}^T$ where $Z = -Y \tan \theta_p$, $\theta$ and $\omega$ are the yaw angle and its velocity. Note that this simplification of state and control input is a special instance of Equation (10).

Robustness to control input noise We recover the ground truth control (thrust force and yaw torque) and its states (location, linear momentum and angular momentum). We add Gaussian noise on the ground truth trajectory to compute the control input force, which is used for initialization. Figure 9 illustrates the estimated trajectory of state and control. The initialization of the control produces erroneous trajectory (blue line). Alternating linearization and policy update in Algorithm 1 allows us to obtain the optimal trajectory (red dotted line) converging to the ground truth trajectory (black line). The noisy control (Figure 9(a)) is also optimized, which produces plausible and smooth control profile aligned with the ground truth. Figure 10(a) shows state reconstruction error as varying the control error where we compare the trajectories 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 10(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 (13). 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 5. Such gravity error can be corrected by force from motion in Algorithm (1), i.e., the trajectory of the state is recovered while the force and torque is computed with respect to the gravity offset. We add gravity noise to the slope, $\theta_p$, to measure the robustness to gravity. In Figure 10(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 $\theta_p$ changes as the gravity error increases.

6.2 Quantitative Evaluation

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 11. 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.

Figure 12(a) shows comparison between initial and optimized states for real mountain biking data. The initialized control input produces implausible profile due to noisy acceleration caused by significant disturbance from environments. The optimal control allows us to find the optimal control input $u$ that minimizes
Gravity error (degree)
Reconstruction error (m)
Initialized state
Optimized state

Fig. 9. We compute the optimal control (a) and its state (b) using Equation (9) 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 (17).

Control input error (Gaussian std.)
Reconstruction error (m)
Initialized state
Optimized state

Fig. 10. 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.

For quantitative evaluation, we design a control experiment with an experienced mountain biker.

Reprojection error, which agrees with structure from motion result as shown in Figure 12(b) and 12(c).

Gravity prediction 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 13(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 13(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 1 summarizes the gravity prediction comparison.

Scale recovery We recover the scale factor and compare the magnitude of linear acceleration with IMU, i.e., $|\mathbf{a}|/|\mathbf{a}_{imu}|$ where $\mathbf{a}$ and $\mathbf{a}_{imu}$ 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 13(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. We use a thresholding binary classifier, $\xi^+(t)$ and $\xi^-(t)$ to detect pedaling and braking, respectively: $\xi^+(t) = 1$ if $\int_{t-\delta t}^t F_T(t)\,dt > \epsilon_T$, and 0 otherwise.

Fig. 11. For quantitative evaluation, we design a control experiment with an experienced mountain biker.

7. Active force and torque are difficult to directly measure using IMU because the measured acceleration is due to net force and torque not input. This requires special force/torque sensors attached human bodies that measures muscle tension.
Fig. 12. (a) Equation (9) produces plausible active force and torque profile that produces a camera trajectory concerting with the video ((b) and (c)).

Fig. 13. (a) We compare our prediction with three baseline algorithms (see the description of the baseline algorithm in Section 6.2. 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).

### Table 1: Gravity prediction error (degree).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y axis</td>
<td>5.62</td>
<td>4.44</td>
<td>4.72</td>
<td>8.10</td>
<td>6.18</td>
<td>9.06</td>
<td>10.15</td>
<td>9.29</td>
<td>6.34</td>
<td>16.02</td>
<td>13.11</td>
<td>10.88</td>
<td>8.31</td>
<td>7.24</td>
<td>5.80</td>
</tr>
<tr>
<td>Y axis MLE</td>
<td>5.92</td>
<td>4.57</td>
<td>5.66</td>
<td>6.08</td>
<td>5.31</td>
<td>5.91</td>
<td>10.68</td>
<td>8.97</td>
<td>9.11</td>
<td>15.83</td>
<td>12.28</td>
<td>11.21</td>
<td>10.09</td>
<td>6.72</td>
<td>8.72</td>
</tr>
<tr>
<td>Ground plane</td>
<td>7.45</td>
<td>6.28</td>
<td>5.14</td>
<td>12.69</td>
<td>8.99</td>
<td>11.31</td>
<td>11.98</td>
<td>10.24</td>
<td>9.03</td>
<td>8.27</td>
<td>5.50</td>
<td>8.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN MLE (ours)</td>
<td>0.76</td>
<td>0.61</td>
<td>0.60</td>
<td>2.53</td>
<td>1.00</td>
<td>4.38</td>
<td>4.40</td>
<td>2.70</td>
<td>3.64</td>
<td>11.21</td>
<td>9.11</td>
<td>5.18</td>
<td>15.17</td>
<td>4.37</td>
<td>4.08</td>
</tr>
</tbody>
</table>

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

Figure 14(a) shows active force profile and ground truth manually annotated from the videos of behavior monitoring cameras as shown in Figure 11. Our active force profile accords with the ground truth, i.e., pedaling when $F_T > 0$ and braking when $F_T < 0$. In Figure 14(b) and 14(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 tasks [27], [51]. Also we compare with Pooled Motion Feature representation [52], 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 camera wearer—the measured acceleration needs to be decomposed.

**Active torque estimation** We compare the estimated angular velocity with measurements from gyroscope in Figure 15. 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 2 summarizes error of angular velocity measured by 11 different scenes. The correlation is also measured, which produces 0.87 mean correlation.

---

8. A sophisticated classifier such as recurrent neural networks can be a complementary approach when supervision is available.
Fig. 14. (a) We identify active forces by manually annotating frames when pedaling or braking. (b) and (c) Our method outperforms optical flow based representation including [52] with a large margin.

Fig. 15. 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.

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

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

6.3 Qualitative Evaluation

We collect 6 categories of sport activities from YouTube: mountain biking (MB), wingsuit flying (WF), skiing (SK), jetskiing at Lake Powell on the Colorado River (JS), speedflying (SF), and urban bike racing in Taxco, Mexico (TX). We extract images and reconstruct scene geometry and camera trajectory in 3D using structure from motion. We assume that all videos have the same intrinsic parameter (fisheye distortion [13], $\omega=0.001619$; focal length, $f_x=547.55$, $f_y=535.48$; principal coordinates, $p_x=640$, $p_y=360$) which we pre-calibrate with GoPro Hero 3 Black edition at the 1280×720 resolution.

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 16 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 camera wearer was banking frequently. For flying case 9, 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.

7 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 (×2 ~ ×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.

We have shown that the inverted pendulum model is simple yet expressive enough to decode dominant physical interactions acting on the camera wearer’s body. This model opens up a new opportunity to understand and analyze human activities and further, will allow to identify what drives such movement.

9. 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.
Fig. 16. 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 physical quantities are illustrated in the next row.

REFERENCES

APPENDIX A
Gravity Training Data

We learn motion and visual cues of the gravitational field from a gravity training data. We employ two methods to address this supervisionary challenges.

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 (computed by structure from motion in Section 4.2), i.e., \( \mathbf{G} \leftrightarrow \{ \mathbf{I}, \mathbf{R} \} \). This gravity annotation scheme is fully automatic, which is scalable.

Uncontrolled data without IMU measurements To reflect the visual semantics of diverse sport activities in Internet first person videos where no 3D gravity annotation is available, we develop a weakly supervised gravity annotation method. We manually label the orientations of a few images of a first person video in 2D, \( \theta_b \leftrightarrow \mathbf{I} \) where the entire video is reconstructed by structure from motion. We combine the 2D gravity annotations and reconstructed 3D orientation of their images to compute the global 3D gravity direction, \( \mathbf{G} \) by minimizing reprojection error of image orientation:

\[
\mathbf{G}(\phi_1^*, \phi_2^*) = \arg \min_{\mathbf{G}(\phi_1, \phi_2)} \sum_{k=1}^{K} \left( \theta_k^b - \tan^{-1} \left( \frac{(\mathbf{R}_k^b)^T \mathbf{G}}{(\mathbf{R}_k^b)^T \mathbf{G}} \right) \right)^2
\]

where \( \theta_k^b \in [-\pi/2, \pi/2] \), \( \mathbf{R}_k^b \) and \( \mathbf{R}_k^b \) are the \( k \)th annotated 2D orientation, first and second rows of the reconstructed rotation matrix, \( \mathbf{R}_k \). Equation (20) is the maximum likelihood estimate of the 3D gravity given annotated images. We solve this by finding global minimum via enumerating discretized \( \phi_1 \in [0, \pi] \) and \( \phi_2 \in [0, 2\pi] \) and refine the solution using a gradient decent optimization. Note that \( \mathbf{G} \) can be ambiguous if \( \{ \mathbf{R}_k \} \) \( k=1 \) undergoes zero yaw angular displacement (rotation about Y axis of the camera) as shown in Figure 4. 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 weakly supervised annotation method allows us to label more than 300,000 frames with less than 2,000 image labeling.

In the training phase, we obtain the ground truth of image orientation label \( \theta_{gt} \) by projecting \( \mathbf{G}_{gt} \) onto each image. We augment the training data by rotating the image about the principal point, \( \mathbf{I}^a = \mathbf{I}(\mathbf{K} \mathbf{M}_{\theta_a} \mathbf{K}^{-1} \mathbf{x}) \) where \( \mathbf{M}_{\theta_a} \in \text{SO}(3) \) is in-plane rotation (rotation about z axis) with \( \theta_a \) 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.

APPENDIX B
Model Validity

We approximate the center of mass with the 3D camera pose. This approximation 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 17(a). Also a similar correlation is observed when the camera wearer undergoes high centrifugal acceleration as shown in Figure 17(b), which validates physical scale recovery from a banked turn in Section 4.4.

APPENDIX C
Inertial Coefficient

We approximate the inertial coefficients, e.g., mass, moment of inertia, and pivot length, based on biomechanical data [37]. 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.
APPENDIX D

ANALYTIC FORM OF JACOBIAN

\[ A_t = \begin{bmatrix} \frac{\partial f_{dyn}}{\partial C} & \frac{\partial f_{dyn}}{\partial P} & \frac{\partial f_{dyn}}{\partial \mathbf{q}} & \frac{\partial f_{dyn}}{\partial \mathbf{L}} \end{bmatrix} \]

\[ \frac{\partial f_{dyn}}{\partial C} = \begin{bmatrix} I_3 \\ 0_{8 \times 3} \end{bmatrix} \]

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

\[ \frac{\partial f_{dyn}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{p \Delta t}{m} \begin{bmatrix} -2q_y & 2q_z & 2q_w & 2q_x \\ 2q_x & 2q_y & 2q_z & 2q_x \\ 0 & -4q_x & -4q_y & 0 \end{bmatrix} \\ \mathbf{I}_3 + \left[ \begin{bmatrix} (J^{-1}L/2)_{11}^T \\ (J^{-1}L/2)_{12}^T \end{bmatrix} \right] \\ \mathbf{I}_3 \otimes (J^{-1}R) \mathbf{R} \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \end{bmatrix} \]

\[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & 0 & -4q_y & -4q_z \\ -2q_z & 2q_y & 2q_x & 2q_y \\ 2q_y & 2q_z & 2q_w & 2q_y \\ 0 & -4q_x & 0 & -4q_z \end{bmatrix} \]

\[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} \\ \Delta t \end{bmatrix} \]

\[ \mathbf{B}_t = \begin{bmatrix} \Delta t \\ 0 \\ 0 \end{bmatrix} \]

where \( \mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T \) and \( \otimes \) is the Kronecker product.

\[ \frac{\partial \hat{x}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial u}{\partial \mathbf{y}} & \frac{\partial v}{\partial \mathbf{y}} & \frac{\partial w}{\partial \mathbf{y}} \end{bmatrix} \]

\[ \frac{\partial}{\partial \mathbf{y}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{C}} & \frac{\partial}{\partial \mathbf{R}} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{C}} & \frac{\partial}{\partial \mathbf{R}} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \]

\[ \frac{\partial}{\partial \mathbf{C}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = -K \mathbf{R} \]

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