Markerless Outdoor Human Motion Capture Using Multiple Autonomous Micro Aerial Vehicles

Nitin Saini[1,*] Eric Price[1] Rahul Tallamraju[1,3] Raffi Enficiaud[2] Roman Ludwig[1]
Igor Martinovic[1] Aamir Ahmad[1] Michael J. Black[1]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany. [2]Pix4D, Berlin, Germany.

[3]Agents and Applied Robotics Group, International Institute of Information Technology, Hyderabad, India.

Figure 1: 3D markerless motion capture from fully autonomous micro aerial vehicles (MAVs) with on-board cameras. Multi-exposure image shows the trajectory of the MAVs and the 3D body pose and shape projected onto an image frame from an external camera. This camera was not part of the motion capture setup. Hence this camera, the MAV's cameras, body pose reprojection and alignment were manually time synchronized.

Abstract

*Capturing human motion in natural scenarios means moving motion capture out of the lab and into the wild. Typical approaches rely on fixed, calibrated, cameras and reflective markers on the body, significantly limiting the motions that can be captured. To make motion capture truly unconstrained, we describe the first fully autonomous outdoor capture system based on flying vehicles. We use multiple micro-aerial-vehicles (MAVs), each equipped with a monocular RGB camera, an **IMU**, and a **GPS** receiver module. These detect the person, optimize their position, and localize themselves approximately. We then develop a markerless motion capture method that is suitable for this challenging scenario with a distant subject, viewed from above, with approximately calibrated and moving cameras. We combine multiple state-of-the-art 2D joint detectors with a 3D human body model and a powerful prior on human pose. We jointly optimize for 3D body pose and camera pose to robustly fit the 2D measurements. To our knowledge, this is the first successful demonstration of outdoor, full-body, markerless motion capture from autonomous flying vehicles.*

*corresponding author email: nitin.saini@tuebingen.mpg.de

1. Introduction

Motion capture is widely used in applications like animation, prosthetics, medical research, robotics, sports, etc. Most of the commercial and widely used motion capture systems are marker based [2]. In these, infrared (IR) reflective markers are placed on the subject's body and tracked using multiple, static, calibrated, IR cameras. This limits the range and naturalness of human motions that can be captured. To increase naturalness, several marker-based systems can be used outside but still require body-mounted

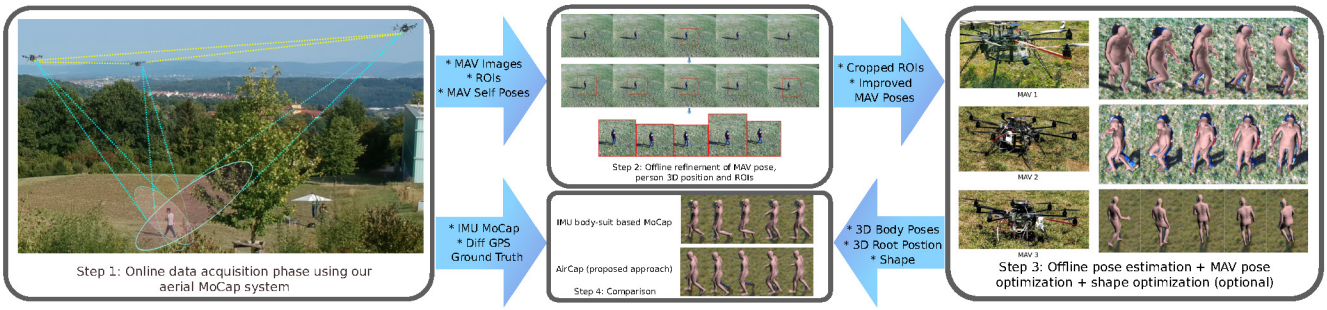

Figure 2: AirCap System Overview. Step 1 is the online phase while Step 2–4 are parts of the offline phase, as described in the text.

markers and fixed, calibrated, cameras. Additionally, markerless systems have been developed for outdoor scenarios but require static cameras [6, 9]. Still, other capture technologies like IMU's can be used [3, 11, 26] but these require wearing bulky sensors, their systems drift over time, and can be influenced by metal objects. Today there is no practical system for outdoor markerless human MoCap that can work over arbitrary distances.

Our goal is to be able to capture a freely moving human outdoors, running, jumping, etc., with no markers and full freedom of movement. To do so, we propose to capture human movement using RGB cameras mounted on multiple micro aerial vehicles (MAVs); that is, a flying motion capture system. This idea is not new but no previous methods achieve our goal. They are either restricted to indoors [31] or require specialized markers to be worn on the body [18].

This problem has remained unsolved because it combines several technologies, each of which on their own is a major technical challenge. First, we need multiple aerial vehicles that can coordinate, self locate, identify a subject, hold them in view, avoid obstacles, etc. Second, motion capture requires calibrated cameras, where the extrinsic parameters are known with high accuracy. Achieving this outdoors with moving vehicles in real settings is a significant technical challenge. Third, while deep learning methods have made 2D joint detection relatively reliable, accurate 3D human pose from images remains an active research problem.

To address these issues, we present an outdoor markerless human motion capture system using a team of MAVs, called *AirCap*. Each MAV consists of only an RGB camera to detect and track the subject. Each MAV also has an onboard IMU, GPS and barometer used only for its self-pose estimation in the global (GPS) coordinate frame. Note that these sensors alone are not sufficient to achieve the accuracy of camera calibration necessary for human MoCap. Consequently, like [7], we formulate the calibration problem together with the human pose estimation problem. The overall functioning of our motion capture system is split into two phases, namely, i) an online data acquisition phase using autonomous MAVs, and ii) an offline pose and shape estimation phase. This is summarized in Fig. 2.

During the online data acquisition phase, the MAVs cooperatively detect and track the 3D position of a subject using the approach presented in [21] and follow them in formation using the perception-driven method from [23]. This formation allows the MAVs to i) keep the subject in their camera's field of view and centered on the image plane, ii) maintain a threshold distance from the subject and iii) not collide with each other or any other static obstacle in the environment. The data acquired in this phase consists of images captured by all MAVs and their camera extrinsic and intrinsic parameters. The data also contains the person's approximate 3D location in the world coordinates (not the detailed pose). Note that the camera extrinsics from this phase are approximate and, as we will see, not sufficiently accurate for human MoCap.

In the second phase, which is offline, human pose and shape are estimated using only the acquired RGB images and the MAV's self-localization poses (the camera extrinsics). Our approach relies on 2D joint detections in each camera; current methods like AlphaPose [8] and OpenPose [5] are quite accurate even with aerial imagery. To fuse these 2D detections into a 3D pose estimate, we formulate an objective function in which we simultaneously solve for body shape, 3D pose, and 3D camera positions. Pose is represented by relative joint rotations of body parts in a kinematic tree. Specifically, we use the 3D SMPL body model [16] to fuse the noisy estimates. SMPL captures the shape of the human body and this constrains the possible solutions. We project the joints of SMPL onto each of the images (using the estimated camera parameters) and compute the error (robustly) between the predictions and the observed 2D detections. Since the 2D pose detections may be noisy, we regularize the 3D fitting with a learned pose prior called Vposer [20]. Vposer is learned from SMPL fits to hours of motion capture data using a variational autoencoder (VAE). We solve for camera parameters jointly and constrain them to be similar to those estimated by the MAVs.

In summary, AirCap addresses the following key challenges: (1) Detection and tracking of a person by

multiple MAVs fully autonomously. (2) Estimation of the camera extrinsics and the 3D location of the person. (3) Fitting a 3D body model robustly to 2D joint detections from multiple flying cameras. (4) We show, for the first time, that it is possible to capture human movement fully autonomously from aerial vehicles. (5) We compare our 3D poses with reference data computed from a multi-IMU suit and the SIP method for pose estimation [26]. While the accuracy is not yet on par with commercial marker-based systems, this is a practical step towards a solution that addresses each piece of technology in an integrated whole. Our code and dataset are available at https://github.com/robot-perception-group/Aircap_Pose_Estimator.

2. Related Work

There is a long history of work on markerless, multi-camera, motion capture. The classical methods all rely on static, calibrated, cameras in laboratory conditions and we do not review these here. Instead, we focus on methods that work outdoors with moving cameras.

Hasler et al. [9] recover human pose from hand-held, unsynchronized, cameras, while more recent work assumes synchronization [28]. Both models assume that there exists a personalized 3D mesh of the person. These methods also assume that the cameras view a scene with a highly textured background that can be used to calibrate the cameras and track their motions using standard structure-from-motion methods. Like [9], Elhayek et al. [7] can deal with unsynchronized cameras, which they time-sync using audio. They also require a 3D template of the body. They require some user interaction to get the initial camera calibration using features and bundle adjustment. They then jointly estimate the body pose and camera calibration parameters. The first use of the body pose for camera calibration was in [14] where they assume a repetitive motion. By using the same pose of the body from different views, they effectively treat the body as a 3D calibration object. In contrast to our scenario, in the above work with hand-held cameras, the human takes up a significant portion of the image. With outdoor aerial applications, the person is frequently far from the camera and the ground may not have sufficient texture for structure from motion.

Flying motion capture systems have primarily been restricted to laboratory environments. Here the vehicles do not need to deal with wind, making the control problem easier. Additionally, indoor environments offer many cues for camera calibration and tracking. For example, the FlyCap system [31] uses RGB-D sensors mounted on multiple indoor micro aerial vehicles (MAVs) [31]. They develop a system for autonomous vehicle control and 3D human pose estimation. However, the method proposed in [31] involves a template scanning as the first step where the subject needs to stay still for some time. FlyCap also requires a textured background for stable flight control. They only test indoors so do not have to deal with wind and fly the drones close to the person so that they are large in the camera field of view.

In contrast, the Flycon system works outdoors but assumes active LED markers are worn on the body. This effectively takes the concept of traditional marker-based mocap, using IR sensors and retroreflective markers, and extends it to flying cameras. Their system works outdoors and the approach leverages the robust and mature algorithms available for IR based MoCap systems. Like earlier work [7] the approach jointly estimates body pose and camera extrinsics. The highly visible markers significantly simplify the problem but require a subject preparation step to place the IR markers on the subject's body. Because of this simplification, Flycon runs in realtime whereas our method takes a two-stage approach. We perform rough realtime 3D tracking of the human during capture and then off-line we estimate the 3D pose. This works well for motion capture but would not be appropriate for a realtime human-robot interaction scenario.

Here we show that explicit LED markers are not necessary, given recent advances in 2D human joint location estimation using deep networks [5, 8]. However, due to the small apparent size of the subject and aerial views, these methods result in a noisy estimate.

Recent work also shows promise in 3D human pose estimation from monocular data [4, 12, 17, 19], but these methods do not use multiple camera views. These 3D estimates from separate cameras cannot be fused easily due to ambiguity in scale and perspectives. In [10] they extend SMPLify to multiple camera views but assume the cameras are stationary. OpenPose [5] also can take multiple calibrated camera images and return 3D joint locations but this approach cannot deal with the inaccurate calibration of flying cameras. Although the 3D estimate from these methods can not be used directly, in our proposed approach we leverage them as noisy sensors for 2D joints positions and show how they can be efficiently fused to obtain a consistent 3D pose and shape estimate.

For outdoor capture, there are other technologies that do not rely on computer vision. Commercial systems, like Xsens are based on subject-mounted inertial measurement units (IMUs) and recent work has shown that body pose can be estimated from a small number of such units [11, 26]. These methods have several limitations however. Subject preparation is required, the subject has to be cooperative, and the sensors can affect movement. Additionally, the IMUs drift and can be significantly affected by metal in the environment. Several methods combine cameras [25, 24] or depth sensors [32] and IMUs to address some of these problems. Here we use an IMU method to create reference data (pseudo ground truth) for the evaluation of our purely RGB

solution.

3. Proposed Approach

We first describe our motion capture hardware and the online phase. Then we discuss our system pipeline in detail by introducing mathematical symbols and notations followed by the algorithm. The pipeline consists of four steps.

3.1. Step 1 : MoCap system setup and online data acquisition phase

Step 1 in Fig. 2 shows our MAV-based outdoor motion capture system tracking and following a person. It consists of a team of self-designed 8-rotor MAVs (see in Step 3 in Fig. 2 inset). Each MAV is equipped with a 2MP HD camera, a computer with an Intel i7 processor, an NVIDIA Jetson TX1 embedded GPU and an OpenPilot Revolution[1] flight controller board. We use the flight controller's position and yaw controller as well as its GPS and IMU-based self-pose (position and orientation) estimation functionalities.

To detect, track and follow the person, we use a perception-driven formation approach [21, 23]. Each copter runs a single shot detector (SSD) multibox [15] on the images acquired by its camera using its on-board GPU to detect the person's outer bounding box on the image frames. A detection rate of $\sim$4 Hz is achieved during the online acquisition. The MAVs then share the person's 2D image bounding box positions and their 3D self-pose estimates wirelessly between each other. Subsequently, using a cooperative detection and tracking (CDT) filter [21] that runs on-board each MAV's CPU, they estimate the 3D position of the person's center of mass in a consistent world frame (GPS-frame). Using this method, the MAVs also improve their 3D self-pose localization. One key feature of the CDT filter is that it allows the detector to focus on the most informative region of interest (ROI) on future image frames, thereby making it computationally efficient. Note that even though the detections are obtained at $\sim$4 Hz, the CDT filter runs at $\sim$30 Hz, alternating between the standard prediction and update steps, except that the updates happen at a lower frequency.

In the online phase, the goal is to keep the person in the field of view and centered in each MAV's camera. Additional constraints include maintaining threshold distances to the other MAVs and static obstacles. To this end, each MAV runs a model predictive control (MPC)-based formation controller [23] on its on-board CPU. The MPC's objective is to maintain a threshold distance to the subject while adhering to the aforementioned formation constrains. Orienting the MAV towards the subject is achieved using an additional yaw controller (separate from the MPC). Further details regarding the CDT tracker and formation controller can be obtained from [21] and [23], respectively.

During the online phase, all MAVs save images on-board at $\sim$40 Hz and their self-pose estimates at $\sim$100 Hz. As the camera is rigidly mounted on each MAV, the extrinsics of the camera are obtained using a fixed and known transformation from the MAV's self-pose (position and orientation) in the world frame.

3.2. Step 2 : 2D region of interest and MAV self pose refinement

In this step, we run the CDT algorithm of Step 1 offline to improve the subject's tracked position estimate and each MAV's self pose estimates. The SSD Multibox detector runs on every frame in Step 2. The CDT filter leverages these every-frame observations to obtain the ROIs for every image and improve the MAV self-pose estimates.

3.3. Step 3 : Offline pose estimation

The rest of this section discusses Step 3 in which the person's pose and shape, as a function of time, is estimated using the data acquired in the online phase (Step 1) and refined in Step 2. Note that Step 4 concerns comparison with ground truth and is, therefore, discussed in the next section with experiments and results.

3.3.1 Preliminaries

Consider a system with C moving cameras. The intrinsic parameters of each camera are fixed. Since the cameras are moving in the world frame, their extrinsic parameters (rotation vector, translation vector) are changing over time. The rotation vector (3×1) and position vector (3×1) of camera c at any time instant t is represented as $\boldsymbol{r}_{c,t}$ and $\boldsymbol{p}_{c,t}$ respectively.

SMPL [16] is a state of the art human body model. It is learned by using thousands of high-quality body scans of people with a wide variety of body types. It is parameterized by two latent parameters: pose and shape. The pose parameter is represented by $\boldsymbol{\theta}$. It is a 72×1 vector, i.e. 3 axis angle values for each of the 23 joints and 3 values for root (pelvis) joint location (23*3+3=72). The SMPL shape parameter $\boldsymbol{\beta}$ is a (10×1) vector whose elements are weights of the 10 most significant eigen shapes (refer [16] for details).

2D joint detections on the collected images can be highly noisy. We use multiple 2D joint detectors for robustness. Say we use D detectors and each detector gives N joints positions on camera plane. The position of n^{th} joint given by d^{th} detector on c^{th} camera plane at time instant t is a 2×1 vector represented as $\boldsymbol{j}_{c,t}^{n,d}$. The detector also gives a confidence value in terms of probability for each detected joint. It is represented as $w_{c,t}^{n,d}$.

[1] OpenPilot: http://www.librepilot.org/site/index.html

The SMPL pose vector $\boldsymbol{\theta}$ is the collection of all the joint angles. However, human poses do not span the entire angle space. To restrict $\boldsymbol{\theta}$ to the natural pose space, we use another parameterization, with a known distribution. This method, also called Vposer, is first introduced in [20]. The new parameterization of human pose, has 32 elements, and is the latent space of a VAE (Variational Auto Encoder) [13] with a Normal distribution. Vposer is trained on more than 1 million poses of multiple subjects and is capable of producing novel, realistic human poses. For more details on the data and actual training procedure refer to [20]. Vposer provides a mapping from the latent variable $\boldsymbol{z}$ to full pose variable $\boldsymbol{\theta}$ given as

$$\boldsymbol{\theta} = \mathcal{V}(\boldsymbol{z}). \tag{1}$$

We can exploit the known distribution of the latent variable as a prior in our optimization objective, by keeping its values close to the mean of the Normal distribution. This translates to a simple L2 norm on the new parameterization.

3.3.2 Algorithm

We use the detected 2D joints and intrinsic parameters to optimize for body model parameters along with camera extrinsics. Camera extrinsics are initialized with the refined estimates obtained in Sec. 3.2. This is done independently for each time step.

Per-frame fitting We minimize a cost function at each time step t, which can be decomposed into the following components:

$$\begin{aligned} E(\boldsymbol{r}_{1\cdots C,t}, \boldsymbol{p}_{1\cdots C,t}, \boldsymbol{z}_t, \boldsymbol{\beta}_t) = \\ E_{2D} + \lambda_{\boldsymbol{r},\boldsymbol{p}} E_{\boldsymbol{r},\boldsymbol{p}} + \lambda_{\boldsymbol{z}} E_{\boldsymbol{z}} + \lambda_{\boldsymbol{\beta}} E_{\boldsymbol{\beta}}, \end{aligned} \tag{2}$$

where $\lambda_{\boldsymbol{r},\boldsymbol{p}}$, $\lambda_{\boldsymbol{z}}$ and $\lambda_{\boldsymbol{\beta}}$ are weights of the corresponding components.

The first term ensures that the 2D projection of the model's 3D joints remains close to the observed 2D joints. It is given as

$$\begin{aligned} &E_{2D}(z_t, \beta_t, \boldsymbol{r}_{c,t}, \boldsymbol{p}_{c,t}) = \\ &\sum_{c,n,d} w_{c,t}^{n,d} \rho_{\sigma_1} \left(\left\| \Pi(\boldsymbol{r}_{c,t}, \boldsymbol{p}_{c,t}, \mathcal{J}^n(\mathcal{V}(\boldsymbol{z}_t), \boldsymbol{\beta}_t)) - \boldsymbol{j}_{c,t}^{n,d} \right\| \right), \end{aligned} \tag{3}$$

where $\mathcal{J}^n$ is the joint regressor function that gives the n^{th} joint position given the SMPL pose and shape parameters. Π is the projection function that projects the 3D point on the image plane, given camera parameters. ρ_{σ_1} is the Geman-McClure robust penalty function with a fixed parameter σ_1, written as

$$\rho_{\sigma_1}(e) = \frac{e^2}{e^2 + \sigma_1^2}. \tag{4}$$

As explained in Sec. 3.1, camera extrinsic parameters are obtained directly from the MAV's self-pose data saved during the flights made by the MAV formation. The self-pose estimates of the MAVs are prone to various sources of error, e.g., GPS and IMU drifts and changing prevelant wind speeds causing fluctuations in the barometer measurements. This causes the camera extrinsic parameters to be noisy. Hence, we also optimize for the camera extrinsic parameters, with the objective of keeping them close to the values estimated online by the MAVs, by including another cost term,

$$E_{\boldsymbol{r},\boldsymbol{p}} = \rho_{\sigma_2}(\boldsymbol{r}_{c,t} - \tilde{\boldsymbol{r}}_{c,t}) + \rho_{\sigma_2}(\boldsymbol{p}_{c,t} - \tilde{\boldsymbol{p}}_{c,t}), \tag{5}$$

where $\tilde{\boldsymbol{r}}_{c,t}$ and $\tilde{\boldsymbol{p}}_{c,t}$ are the rotation and position vectors of camera c at any time t estimated online by the MAVs during the data acquisition phase. ρ_{σ_2} is the same function described in (4).

$E_{\boldsymbol{z}}$ is a regularization term on the latent pose parameter $\boldsymbol{z}$ given as

$$E_{\boldsymbol{z}} = \|\boldsymbol{z}\|. \tag{6}$$

$\boldsymbol{\beta}$ is a vector of the 10 most significant eigen shapes of SMPL, which we regularize with $E_{\boldsymbol{\beta}}$ as

$$E_{\boldsymbol{\beta}} = \|\boldsymbol{\beta}\|. \tag{7}$$

4. Experiments and Results

4.1. Data Acquisition

Using our MAV-based motion capture system described in Sec. 3.1, we performed a data collection formation flight using 3 MAVs. Our on-board formation controller, MAV self-pose and person's (3D position, not joint poses) state estimator, etc., are implemented as Robot Operating System (ROS) nodes which makes it easier for MAVs to communicate with each other using standard message types. The MAV formation constraints of altitude and horizontal distance from the subject is set to 8m. The value is relatively high due to safety considerations. During the formation flight, the subject is requested to walk on a grassy field at slow to moderate speeds and later perform random motion sequences, such as jumping jacks, bending forward/backward, swaying arms, etc.

4.2. Dataset

All images and camera extrinsic and intrinsic parameters are saved on-board each MAV as ROS messages in a rosbag file. Each message has Unix timestamp denoting the time of its acquisition. We receive images from each camera at approx. 30-40 frames per second (fps). Even though both

MAV cameras have the same frame rates, they are not synchronized. Meaning, they do not necessarily capture image frames simultaneously. For any image from the first MAV's camera, there might not exist an image from the other MAV's camera at the same instant. Also, as camera parameters are available at a much higher frequency than images, for each image in the system, camera intrinsic and extrinsic parameters are available. Later, we extract data from the saved bagfile, refine them and use it to estimate the shape and pose of the subject using the method described in Sec. 3.3.

4.3. Reference Data

We obtain reference (ref) data to evaluate our reconstructions from two different systems, i) a commercially available IMU MoCap system (Xsens) [3] and ii) a pair of differential GPS modules. IMU system is used to obtain reference data for body pose relative to the root joint. For reference SMPL parameters, we use a state of the art IMU MoCap method Sparse Inertial Poser (SIP) [27]. It uses raw data from Xsens and gives SMPL parameters. However, the global root joint position and orientation from SIP are not reliable for ref comparison. To solve this issue, we use a pair of differential GPS modules, each one attached to a shoulder of the subject to get the position of root joint in the global coordinate system. The reference global root orientation still remains unestimated as it is not directly measurable with these two systems.

4.4. Implementation

Using the approach in [21] the MAVs autonomously maintain a formation around the person while following him/her and keeping him/her centered in their camera's field of view. During the formation flights, the MAVs detect the person in their camera image using single shot detector (SSD) multibox [15] and estimate his/her 3D world position (not the joint pose) and uncertainty associated, in order to maintain the formation. This also results in a cropped region of interest (ROI) which has the highest likelihood of having the person inside it. For every image, the MAVs also save this corresponding ROI. The ROI data and MAV self pose estimates are then refined offline and saved. We crop the full images based on the provided ROIs and apply multiple joint detectors, each producing a set of 2D joints estimates. If the ROI goes outside the camera frame, we take the full image for 2D joint detection.

We then use two state of the art 2D joint detectors: alphapose [8, 30] and OpenPose [5, 22, 29]. All the dataset images are processed using these joint estimators and their output is saved with the same timestamp as that of the image. We use these 2D joints along with the camera extrinsic and intrinsic parameters in our cost function as given in (2). Since the cameras are not synchronized, we use the closest frames in time from all the cameras for per-frame fitting. We use a PyTorch [1] implementation of SMPL to regress from SMPL parameters to 3D joint positions in (3). The total cost is sequentially minimized for each frame to get the optimized value of SMPL pose and camera extrinsic parameters. The value of σ_1 (3) and σ_2 (5) are 40 and 10 respectively. We found after trial-and-error that these values work well. After optimizing for a frame, the optimized parameter values are used as initial values for the next frame except for the camera extrinsics. These are initialized with the ones obtained from Sec 3.2. For optimization, we use the Adam optimizer from PyTorch. The number of iterations for the first frame is 1000 with 0.25 learning rate and 100 with 0.1 learning rate for subsequent frames.

	actual shape		shape estimation	
Joint	e_{jp}	e_{ja}	e_{jp}	e_{ja}
L_Hip	0	6.73	0	6.81
L_Knee	0.0767	8.60	0.0876	8.69
L_Ankle	0.1629	5.49	0.1904	5.50
L_Foot	0.1843	9.71	0.2157	9.44
R_Hip	0	6.62	0	6.60
R_Knee	0.0680	9.59	0.0760	9.67
R_Ankle	0.1251	7.79	0.1448	7.73
R_Foot	0.1461	8.10	0.1693	7.86
Spine1	0	5.32	0	5.18
Spine2	0.0264	3.01	0.0290	2.96
Spine3	0.0397	1.61	0.0439	1.59
Neck	0.0931	6.25	0.1068	6.11
Head	0.1237	5.13	0.1428	4.90
L_Collar	0.0683	4.09	0.0771	3.82
L_Shoulder	0.0779	13.15	0.0861	13.28
L_Elbow	0.0863	16.41	0.1023	16.15
L_Wrist	0.1689	10.46	0.1984	10.21
L_Hand	0.2045	2.34	0.2411	2.30
R_Collar	0.0694	5.55	0.0777	5.23
R_Shoulder	0.0919	10.96	0.0993	10.87
R_Elbow	0.0987	22.15	0.1075	21.65
R_Wrist	0.1781	11.41	0.2013	11.29
R_Hand	0.2134	3.19	0.2417	3.19
Pelvis	aligned with the ref			

Table 1: Mean error in joint positions (meters) and joint angles (degrees) (using actual body shape vs shape estimation). Pelvis joint is aligned with the ref. The position error for L_Hip, R_Hip and Spine1 becomes 0 because these joints are rigidly connected to the Pelvis.

4.5. Results and Discussion

First, we compare our reconstructed pose with the reference pose. In this, we zero out the global position and rotation of the reconstructed SMPL and ref SMPL. In Table 1, we show the mean error in joint positions (e_{jp}) and mean error in joint angles (e_{ja}). e_{jp} is calculated by taking the Euclidean distance between each estimated joint and the

Figure 3: Pose and shape estimation results of our approach overlaid on some of the image sequences from one of the MAV's camera. (Left) A walking sequence. (Right) A sequence with arbitrary hand and leg movement.

corresponding reference joint and then calculating its mean over the whole dataset. e_{ja} is calculated by taking angle difference between the reconstructed joint angle and the corresponding reference joint angle and taking a mean over time and over all the 3 axes. We show these errors in two cases 1) using actual body shape 2) with subject shape estimation. In case 1, we fix the shape to the actual shape obtained by scanning the person. In case 2, the shape is optimized in Step 3 of the system pipeline (2). We observe that the error is higher for the joints corresponding to the extremities. This is because vposer is not trained with the extreme poses. We use a pose regularization in (6) which penalizes the distance from the mean pose of vposer. Since, the extreme poses have more variation in extremity joints, these joints are penalized more. We notice that including shape estimation in Step 3 does not affect the error significantly. However, the per-frame fitting does not make sure that the shape remains same for the whole sequence. For a better shape estimate, instead of the per-frame fitting, the complete sequence should be optimized with constant shape for the whole sequence.

For all the further results, we fix the shape to the actual shape of the person and all the errors presented are joint position errors in meters.

4.5.1 Pose Evaluation

In Fig. 3 we present qualitative results of our pose estimation. Quantitative results are presented in Fig. 4, where we show the mean joint position error with time. At every time instant we calculate the *number of detections* as $\max_d \sum_c \mathcal{D}_c^d$, where $\mathcal{D}_c^d$ denotes the detections obtained by detector d for MAV camera c. In Fig. 4 we show the *number of detections* in the background represented by a color scheme. We can see that the mean joint position error becomes high when the *number of detections* is less, which was expected from our approach. This shows that observations from multiple views add confidence to the estimated pose. For more images and renderings of the experiments see supplementary material.

4.5.2 Global Position Evaluation

We use the absolute position from the differential GPS modules mounted on the subject's shoulders. These modules are

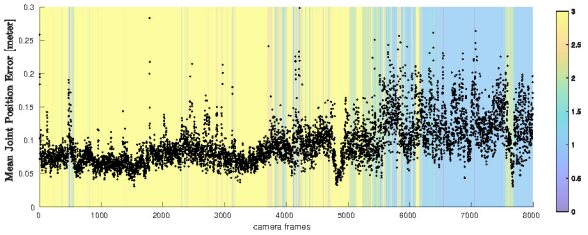

Figure 4: Mean joint position error in every frame. Background color denotes the number of detections at that frame calculated as explained in Sec.4.5.1.

in a coordinate system which has a constant offset to our system's coordinate system. We find this offset by taking a mean difference between estimated and ref position for first 200 frames and correct it manually.

All the Steps 1, 2 and 3 provide the person's position. We denote these by $\mathcal{J}^0_{S1}$,$\mathcal{J}^0_{S2}$ and $\mathcal{J}^0_{S3}$ respectively and the ref root position by $\mathcal{J}^0_{ref}$. We show the X, Y and Z components of these in Fig. 5. In the motion sequence shown in this experiment, the subject first moves on a zig-zag trajectory over a sloped terrain with moderate speed. This can be seen in the ref plots for the initial 4700 frames. Then the subject performs various random body pose sequences like jumping jacks, punching, dancing etc., with small motion in global position. This is reflected in the plots as there is not much variability in the ref position.

In the inset of Fig.5, we show box plot of the Euclidean error of $\mathcal{J}^0_{S1}$, $\mathcal{J}^0_{S2}$, $\mathcal{J}^0_{S3}$ with respect to $\mathcal{J}^0_{ref}$. We see that the estimate improves in Step 2 and further in Step 3. If we do not optimize for camera parameters in Step 3 our person position estimates are unchanged from that of Step 2. This indicates that the optimization of camera parameters improves the person's global pose estimate. However, looking at the outliers we can say that the maximum error can go even higher than the maximum error of Step 1. To analyze this, we look at the last three plots of Fig.5. The background represents the same as in Fig.4. In these plots, we show the signed error of $\mathcal{J}^0_{S1}$, $\mathcal{J}^0_{S2}$, $\mathcal{J}^0_{S3}$ with respect to $\mathcal{J}^0_{ref}$. Notice there are two error components in these plots. One is a slowly varying component and another is rapidly varying. We show the rapidly varying component by plotting a moving average result over the error. The slow varying error is due to the drift in MAVs GPS. Since the person's position estimate is dependent on the MAVs poses, this drift is reflected in the person's position error. The rapidly moving

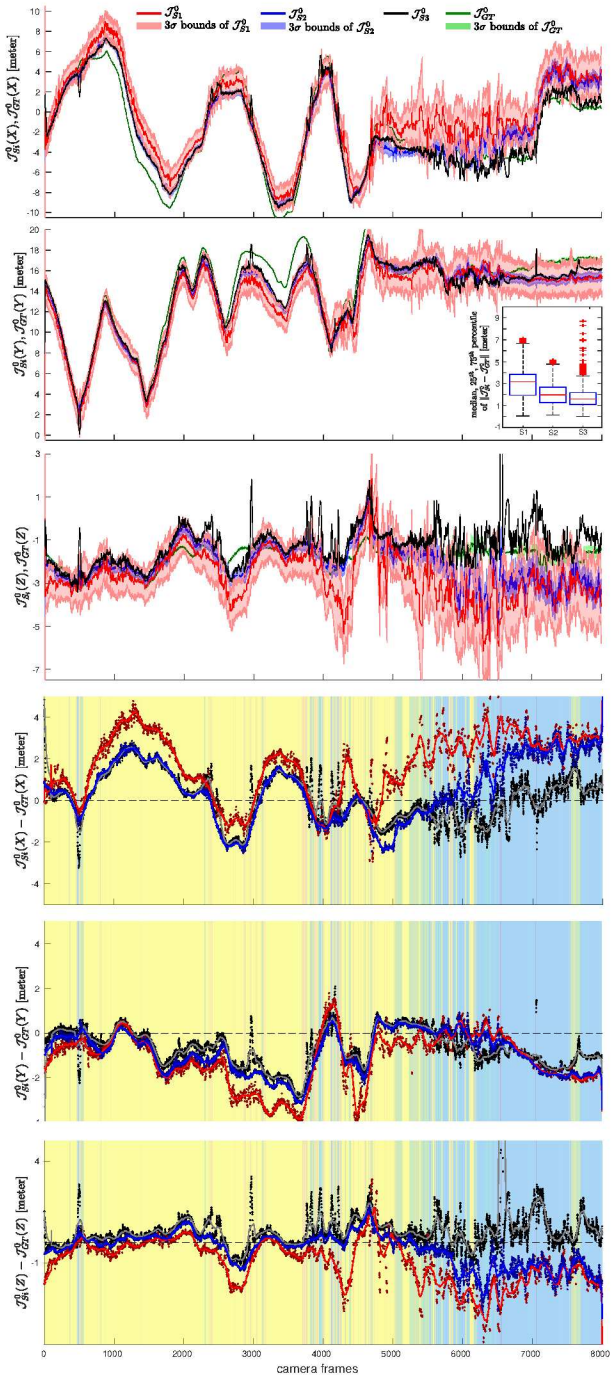

Figure 5: Top three plots show the root position trajectories from ref, Step 1,2 and 3 in X, Y and Z dimension, respectively. Bottom three plots are signed error of the estimated root position from all three steps with respect to the ref. See Sec.4.5.2 for details.

error is due to the observation error in our 2D estimates. We see there are sudden jumps in the pose error from Step 3. These correspond to the outliers shown in the inset of the second plot of Fig.5. We see in the last three plots that these jumps happen when there are fewer detections, which is expected. Since there are fewer detections or no detections in some camera frames, the whole optimization becomes unconstrained. In such a scenario, it becomes highly susceptible to observation errors and the camera pose can be adjusted to fit an erroneous observation.

5. Conclusion

In this paper, we presented AirCap, the first successful demonstration of full-body markerless motion capture from autonomous flying vehicles. AirCap addresses the challenges of i) online image data acquisition of a tracked human subject by multiple fully autonomous MAVs, and ii) human body pose and shape estimation using the acquired image dataset. We show how we leverage state-of-the-art 2D human joint detection methods as noisy sensors and fuse them to obtain consistent 3D estimates of human pose and shape. We show quantitative results by evaluating our reconstructions using reference data. We also show qualitative results by projecting the estimated pose over the acquired images. One of the most important advantages of our method is that it completely removes the need for a subject preparation step, thereby allowing in-the-wild motion capture of any subject.

6. Limitations and Future Work

The main limitation of our approach is the estimate of the root joint position. Since we compute camera extrinsics estimates from Step 2 and do not optimize over them in a robust way, we are limited by them. Another limitation of our system is that, as the MAV navigates, it could lose the subject from its FOV during transient behavior. Since the camera is rigidly attached to the MAVs and our convex MPC is formulated with the assumption of linear MAV dynamics, a MAV's camera, and hence the image it acquires, appears to shake when the MAV changes its motion direction. In the future, we plan to address this by mounting the camera on a gimbal attached to the MAV frame and separately handle its control. We plan to explore methodologies to optimize camera extrinsics in an integrated approach rather than doing it in three hierarchical steps. We also want to utilize realistic human motion models to improve the temporal naturalness of the captured motion. Finally, extending our method to larger and complex outdoor scenarios as well as motion capturing multiple subjects are also included in our future work.

Acknowledgements: We thank our colleagues Nima Ghorbani and Vassilis Choutas for providing early implementation of VPoser and PyTorch SMPL.

Disclosure: MJB has received research gift funds from Intel, Nvidia, Adobe, Facebook, and Amazon. While MJB is a part-time employee of Amazon, his research was performed solely at, and funded solely by, MPI. MJB has financial interests in Amazon and Meshcapade GmbH

References

[1] Pytorch. https://pytorch.org/.

[2] Vicon motion capture system. www.vicon.com.

[3] Xsens motion capture system. https://www.xsens.com.

[4] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Mykhaylo Andriluka, Christoph Bregler, Bernt Schiele, and Christian Theobalt. Marconi - convnet-based marker-less motion capture in outdoor and indoor scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):501–514, 2017.

[7] Ahmed Elhayek, Carsten Stoll, Kwang In Kim, and Christian Theobalt. Outdoor human motion capture by simultaneous optimization of pose and camera parameters. *Comput. Graph. Forum*, 34(6):86–98, Sept. 2015.

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[9] Nils Hasler, Bodo Rosenhahn, Thorsten Thormählen, Michael Wand, Juergen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 224–231, 2009.

[10] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Javier Romero, Ijaz Akhter, and Michael J. Black. Towards accurate marker-less human shape and pose estimation over time. In *International Conference on 3D Vision (3DV)*, 2017.

[11] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37:185:1–185:15, Nov. 2018. Two first authors contributed equally.

[12] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018.

[13] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes.

[14] David Liebowitz and Stefan Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. In *ICCV*, 2001.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[16] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[17] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. volume 36, July 2017.

[18] Tobias Nägeli, Samuel Oberholzer, Silvan Plüss, Javier Alonso-Mora, and Otmar Hilliges. Real-time environment-independent multi-view human pose estimation with aerial vehicles. 2018.

[19] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, Sept. 2018.

[20] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Eric Price, Guilherme Lawless, Roman Ludwig, Igor Martinovic, Heinrich H. Bülthoff, Michael J. Black, and Aamir Ahmad. Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles. *IEEE Robotics and Automation Letters*, 3(4):3193–3200, Oct. 2018. Also accepted and presented in the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[22] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[23] Rahul Tallamraju, Eric Price, Roman Ludwig, Kamalakar Karlapalem, Heinrich H Bülthoff, Michael J Black, and Aamir Ahmad. Active perception based formation control for multiple aerial vehicles. *IEEE Robotics and Automation Letters*, pages 1–1, 2019.

[24] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *European Conference on Computer Vision (ECCV)*, Sept. 2018.

[25] Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. Human pose estimation from video and imus. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, jan 2016.

[26] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 349–360, 2017.

[27] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 349–360, 2017.

[28] Yangang Wang, Yebin Liu, X. Tong, Qionghai Dai, and Ping Tan. Outdoor markerless motion capture with sparse handheld video cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24:1856–1866, 2018.

[29] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[30] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.

[31] Lan Xu, Yebin Liu, Wei Cherng Cheng, Kaiwen Guo, Guyue Zhou, Qionghai Dai, and Lu Fang. Flycap: Markerless motion capture using multiple autonomous flying cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24:2284–2297, 2016.

[32] Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu. HybridFusion: Real-time performance capture using a single depth sensor and sparse IMUs. 2018.