ActiveMoCap: Optimized Viewpoint Selection for Active Human Motion Capture

Sena Kiciroglu[1] Helge Rhodin[1,2] Sudipta N. Sinha[3] Mathieu Salzmann[1] Pascal Fua[1]
[1] CVLAB, EPFL [2] Imager Lab, UBC [3] Microsoft

Abstract

The accuracy of monocular 3D human pose estimation depends on the viewpoint from which the image is captured. While freely moving cameras, such as on drones, provide control over this viewpoint, automatically positioning them at the location which will yield the highest accuracy remains an open problem. This is the problem that we address in this paper. Specifically, given a short video sequence, we introduce an algorithm that predicts which viewpoints should be chosen to capture future frames so as to maximize 3D human pose estimation accuracy. The key idea underlying our approach is a method to estimate the uncertainty of the 3D body pose estimates. We integrate several sources of uncertainty, originating from deep learning based regressors and temporal smoothness. Our motion planner yields improved 3D body pose estimates and outperforms or matches existing ones that are based on person following and orbiting.

1. Introduction

Monocular approaches for 3D human pose estimation have improved significantly in recent years, but their accuracy remains relatively low. In this paper, we explore the use of a moving camera whose motion we can control to resolve ambiguities inherent to monocular 3D reconstruction and to increase pose estimation accuracy. This is known as *active vision* and has received surprisingly little attention in the context of using modern approaches to body pose estimation. An active motion capture system, such as one based on a personal drone, would allow one to film themselves performing a physical activity and analyze their motion, for example to get feedback on their performance. When using only one camera, the quality of such feedback will strongly depend on selecting the most beneficial viewpoints for pose estimation. Fig. 1 depicts an overview of our approach based on a drone-based monocular camera.

In this paper, we introduce an algorithm designed to continuously position a moving camera at optimal viewpoints to maximize the 3D pose estimation accuracy for a freely moving subject. We achieve this by moving the camera in 6D pose space to viewpoints that maximize a utility function designed to predict reconstruction accuracy. However, the utility function cannot be defined in terms of reconstruction accuracy because doing so would require knowing the true person and camera position, leading to a chicken and egg problem. Instead we use prediction uncertainty as a surrogate for accuracy. This is a common strategy used in robot navigation systems for unknown scenes where the robot explores areas that are most incomplete in its internal map representation [20]. However, in our situation, estimating uncertainty is much more difficult since multiple sources of uncertainty need to be considered. These include uncertainties about what the subject will do next, the reliability of the pose estimation algorithm, and the accuracy of distance estimation along the camera's line of sight.

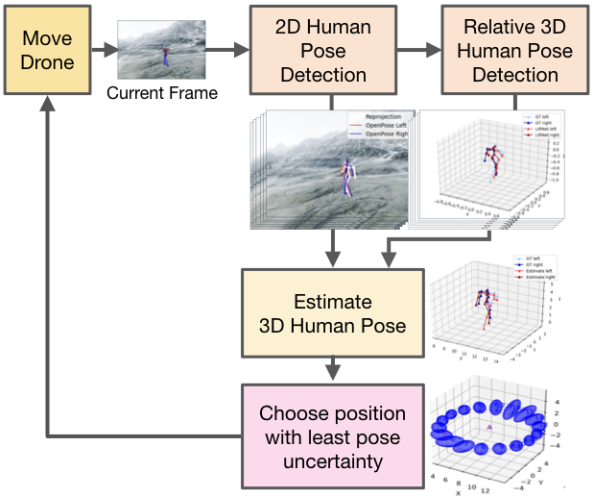

Figure 1. **Method overview.** The 2D and 3D human pose is inferred from the current frame of the drone footage, using off the shelf CNNs. The 2D pose and relative 3D pose of the last k frames is then used to optimize for the global 3D human motion. The next view of the drone is chosen so that the uncertainty of the human pose estimation from that view is minimized, which improves reconstruction accuracy.

Our key contribution is therefore a formal model that provides an estimate of the *posterior variance* and probabilistically fuses these sources of uncertainty with appropriate prior distributions. This has enabled us to develop an active motion capture technique that takes raw video footage as input from a moving aerial camera and continuously computes future target viewpoints for positioning the camera, in a way that is optimized for human motion capture. We demonstrate our algorithm in two different scenarios and compare it against standard heuristics, such as constantly rotating around the subject and maintaining a constant angle with respect to the subject. We find that when allowed to choose the next viewpoint without physical constraints, our algorithm outperforms the baselines consistently. For simulated drone flight, our results are on par with constant rotation, which we conclude is the best trajectory to choose in the case of no obstacles blocking the circular flight path. Our code is available at `https://github.com/senakicir/ActiveMoCap`

2. Related work

Most recent approaches to markerless motion capture rely on deep networks that regress 3D pose from monocular images [16, 17, 21, 38, 25, 31, 22, 44, 36, 34, 41, 39, 15]. While a few of these methods improve robustness by enforcing temporal consistency [23], none considers the effect that actively controlling the camera may have on accuracy. The methods most closely related to ours are therefore those that optimize camera placement in multi-camera setups and those that guide robots in a previously-unknown environment.

Optimal Camera Placement for Motion Capture. Optimal camera placement is a well-studied problem in the context of static multi-view setups. Existing solutions rely on maximizing image resolution while minimizing self-occlusion of body parts [5, 2] or target point occlusion and triangulation errors [27]. However, these methods operate offline and on pre-recorded exemplar motions. This makes them unsuitable for motion capture using a single moving camera that films *a priori* unknown motions in a much larger scene where estimation noise can be high.

In [24] multiple cameras poses are optimized for triangulation of joints in a dome environment using a self-supervised reinforcement learning approach. In our case, we consider the monocular problem. Our method is not learning based, we try to obtain the next best view from the loss function itself.

View Planning for Static and People Reconstruction. There has been much robotics work on active reconstruction and view planning. This usually involves moving so as to maximize information gain while minimizing motion cost, for example by a discretizing space into a volumetric grid and counting previously unseen voxels [14, 8] or by accumulating estimation uncertainty [20]. When a coarse scene model is available, an optimal trajectory can be found using offline optimization [30, 13]. This has also been done to achieve desired aesthetic properties in cinematography [11]. Another approach is to use reinforcement learning to define policies [7] or to learn a metric [12] for later online path planning. These methods deal with rigid unchanging scenes, except the one in [6] that performs volumetric scanning of people during information gain maximization. However, this approach can only deal with very slowly moving people who stay where they are.

Human Motion Capture on Drones. Drones can be viewed as flying cameras and are therefore natural targets for our approach. One problem, however, is that the drone must keep the person in its field of view. To achieve this, the algorithm of [45] uses 2D human pose estimation in a monocular video and non-rigid structure from motion to reconstruct the articulated 3D pose of a subject, while that of [18] reacts online to the subject's motion to keep them in view and to optimize for screen-space framing objectives. AirCap [32] calculates trajectories of multiple drones that aim to keep the person in view while simultaneously performing object avoidance. This was extended in [35] so as to optimize multiple MAV trajectories by minimizing the uncertainty of the 3D human joint positions being tracked, but focusing on the 3D human pose estimation as an offline step. In [19], this was integrated into an autonomous system that actively directs a swarm of drones and simultaneously reconstructs 3D human and drone poses from onboard cameras. This strategy implements a pre-defined policy to stay at constant distance to the subject and uses pre-defined view angles (90° between two drones) to maximize triangulation accuracy. This enables mobile large-scale motion capture, but relies on markers for accurate 2D pose estimation. In [40], three drones are used for markerless motion capture, using an RGBD video input for tracking the subject.

In short, existing methods either optimize for drone placement but for mostly rigid scenes, or estimate 3D human pose but without optimizing the camera placement. [24] performs optimal camera placement for multiple cameras. Here, we propose an approach that aims to find the best next drone location for monocular view so as to maximize 3D human pose estimation accuracy.

3. Active Human Motion Capture

Our goal is to continuously position the camera in 6D pose space so that the acquired by the camera can be used to achieve the best overall human pose estimation accuracy. What makes this problem challenging is that, when we decide where to send the camera, we do not yet know where

the subject will be and in what position exactly. We therefore have to guess. To this end, we propose the following three-step approach depicted by Fig. 1:

1. Estimate the 3D pose up to the current time instant.
2. Predict the person's future location and 3D pose at the time the camera acquires the next image, including an uncertainty estimate.
3. Select the optimal camera pose based on the uncertainty estimate and move the camera to that viewpoint.

We will consider two ways the camera can move. In the first case, the camera can teleport from one location to the next without restriction, allowing us to explore the theoretical limits of our approach. Such a teleportation mode can be simulated using a multi-camera setup, enabling us to evaluate our model on both simulated data and real image datasets acquired from multiple viewpoints. In the second, more realistic scenario, the camera is carried by a simulated drone, and we must take into account physical limits about the motion it can undertake.

3.1. 3D Pose Estimation

The 3D pose estimation step takes as input the video feed from the on-board camera over the past N frames and outputs for each frame, $t \in (1, \ldots, N)$, the 3D human pose, represented as 15 3D points $\mathbf{\Theta}^t \in \mathbb{R}^{15\times 3}$, and the drone pose, as 3D position and rotation angles $\mathbf{D}^t \in \mathbb{R}^{2\times 3}$. Our focus is on estimating the 3D human pose using the real-time method proposed by [3], which detects the 2D locations of the human's major joints in the image plane, $\mathbf{M}^t \in \mathbb{R}^{15\times 2}$, and the subsequent use of [36], which lifts these 2D predictions to 3D pose, $\mathbf{L}^t \in \mathbb{R}^{15\times 3}$. However, these per-frame estimates are error prone and relative to the camera. To remedy this, we fuse 2D and 3D predictions with temporal smoothness and bone-length constraints in a space-time optimization. This exploits the fact that the drone is constantly moving so as to disambiguate the individual estimates. The bone lengths, $\mathbf{b}_{\text{calib}}$, of the subject's skeleton are computed during an apriori calibration stage, where the subject has to stand still for 20 seconds. This is performed only once for each subject. Formally, we optimize for the global 3D human pose by minimizing an objective function E_{pose}, which we detail below.

3.1.1 Formulation

Our primary goal is to improve the global 3D human pose estimation of a subject changing position and pose. We optimize the time-varying pose trajectories across the last k frames. Let t be the last observed frame. We capture the trajectory of poses $\mathbf{\Theta}^{t-k}$ to $\mathbf{\Theta}^t$ in the pose matrix $\mathbf{\Theta}$. We then write an energy function

$$E_{\text{pose}} = E_{\text{proj}}(\mathbf{\Theta}, \mathbf{M}, \mathbf{D}) + E_{\text{lift}}(\mathbf{\Theta}, \mathbf{L}) + E_{\text{smooth}}(\mathbf{\Theta}) + E_{\text{bone}}(\mathbf{\Theta}, \mathbf{b}) \; . \tag{1}$$

The individual terms are defined as follows. The lift term, E_{lift}, leverages the 3D pose estimates, $\mathbf{L}$, from LiftNet [36]. Because these are relative to the hip and without absolute scale, we subtract the hip position from our absolute 3D pose, $\mathbf{\Theta}^t$, and apply a scale factor m to $\mathbf{L}$ to match the bone lengths $\mathbf{b}_{\text{calib}}$ in the least-square sense. We write

$$E_{\text{lift}}(\mathbf{\Theta}, \mathbf{L}) = \omega_l \sum_{i=t-k}^{t} \| m \cdot \mathbf{L}^i - (\mathbf{\Theta}^i - \mathbf{\Theta}^i_{\text{hip joint}}) \|_2^2 \; , \tag{2}$$

with ω_l its relative weight.

The projection term measures the difference between the detected 2D joint locations and the projection of the estimated 3D pose in the least-square sense. We write it as

$$E_{\text{proj}}(\mathbf{\Theta}, \mathbf{M}, \mathbf{D}) = \omega_p \sum_{i=t-k}^{t} \| \mathbf{M}^i - \Pi(\mathbf{\Theta}^i, \mathbf{D}^i, \mathbf{K}) \|_2^2 \; , \tag{3}$$

where Π is the perspective projection function, $\mathbf{K}$ is the matrix of camera intrinsic parameters, and ω_p is a weight that controls the influence of this term.

The smoothness term exploits that we are using a continuous video feed and that the motion is smooth by penalizing velocity computed by finite differences as

$$E_{\text{smooth}}(\mathbf{\Theta}) = \omega_s \sum_{i=t-k+1}^{t} \| (\mathbf{\Theta}^{i+1} - \mathbf{\Theta}^i) \|_2^2 \; . \tag{4}$$

with ω_s as its weight.

To further constrain the solution space, we use our knowledge of the bone lengths $\mathbf{b}_{\text{calib}}$ found during calibration and penalize deviations in length. The length of each bone b in the set of all bones b_{all} is found as $\mathbf{b}_b^t = \| (\mathbf{\Theta}_{b_1} - \mathbf{\Theta}_{b_2}) \|_2$ for frame t. The bone length term is then defined as

$$E_{\text{bone}}(\mathbf{\Theta}) = \omega_b \sum_{i=t-k}^{t} \sum_{b \in b_{\text{all}}} d(\mathbf{b}_b^i, \mathbf{b}_{\text{calib},b}) \; , \tag{5}$$

with ω_b as its weight.

The complete energy E_{pose} is minimized by gradient descent at the beginning of each control cycle, to get a pose estimate for control. The resulting pose estimate $\hat{\mathbf{\Theta}}$ is the maximum a posteriori estimate in a probabilistic view.

3.1.2 Calibration Mode

Calibration mode only has to be run once for each subject to find the bone lengths, $\mathbf{b}_{\text{calib}}$. In this mode, the subject is assumed to be stationary. The situation is equivalent to having the scene observed from multiple stationary cameras, such as in [29]. We find the single static pose $\mathbf{\Theta}^{\text{c}}$ that minimizes

$$E_{\text{calib}} = E_{\text{proj}}(\mathbf{\Theta}^{\text{c}}, \mathbf{M}, \mathbf{D}) + E_{\text{symmetry}}(\mathbf{\Theta}^{\text{c}}). \tag{6}$$

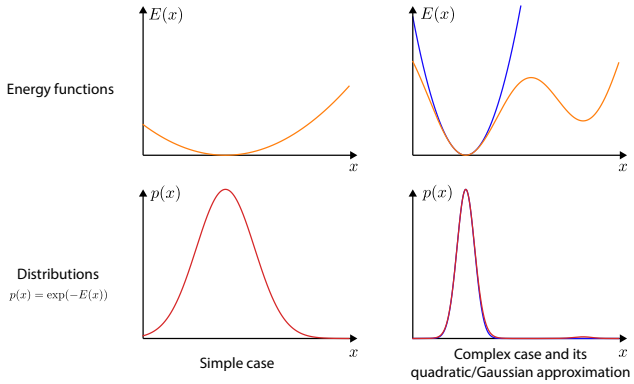

Figure 2. **Probabilistic interpretation.** Left: A quadratic energy function and its associated Gaussian error distribution. Right: A complex energy function, which is locally approximated with a Gaussian (blue) near the minimum. The curvature of the energy function is a measure of the confidence in the estimate and the variance of the associated error distribution. The energy on the right is more constrained and its error distribution has a lower variance.

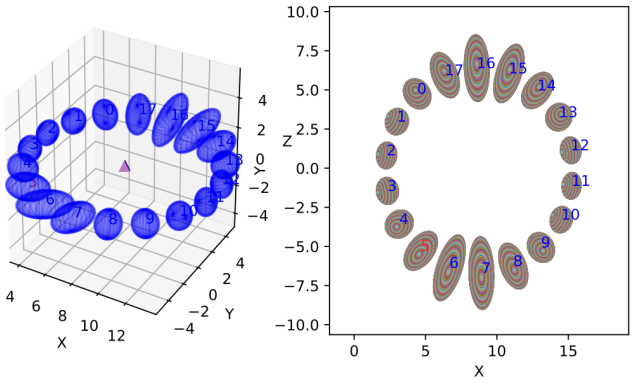

Figure 3. **Uncertainty estimates** for each candidate drone position, visualized on the left as 3D ellipsoids and on the right from a 2D top-down view. Each ellipse visualizes the eigenvalues of the hip location when incorporating an additional view from its displayed position. Here, the previous image was taken from the top (position 16) and uncertainty is minimized by moving to an orthogonal view. The complete distribution has more than three eigenvectors and cannot straightforwardly be visualized in 3D.

In this objective, the projection term, E_{proj}, is akin to the one in our main formulation but acts on all calibration frames. It can be written as

$$E_{\text{proj}}(\boldsymbol{\Theta}^{\text{c}}, \mathbf{M}, \mathbf{D}) = \omega_{\text{p}} \sum_{i=0}^{t} \|\mathbf{M}^i - \Pi(\boldsymbol{\Theta}^{\text{c}}, \mathbf{D}^i, \mathbf{K})\|_2^2 \ , \quad (7)$$

with ω_{p} controlling its influence. The symmetry term, E_{symmetry}, ensures that the left and right limbs of the estimated skeleton have the same lengths by penalizing the squared difference of their lengths.

3.2. Next Best View Selection

Our goal is to find the next best view for the drone at the future time step $t+1$, $\mathbf{D}^{t+1}$. We will model the uncertainty of the pose estimate in a probabilistic setting. Let $p(\boldsymbol{\Theta}|\mathbf{M}, \mathbf{D}, \mathbf{L}, \mathbf{b})$ be the posterior distribution of poses. Then, E_{pose} is its negative logarithm and its minimization corresponds to maximum a posteriori (MAP) estimation. In this formalism, the sum of the individual terms in E_{pose} models that our posterior distribution is composed of independent likelihood and prior distributions. For a purely quadratic term, $E(x) = \omega(x-\mu)^2$, the corresponding distribution $p_E = \exp(-E)$ is a Gaussian with mean μ and standard deviation $\sigma = \frac{1}{\sqrt{2\omega}}$. Notably, σ is directly linked to the weight ω of the energy. Most of our energy terms involve non-linear operations, such as perspective projection in E_{proj}, and therefore induce non-Gaussian distributions, as visualized in Fig. 2. Nevertheless, as for the simple quadratic case, the weights ω_p and ω_l of E_{proj} and E_{lift} can be interpreted as surrogates for the amount of measurement noise in the 2D and 3D pose estimates.

A good measure of uncertainty is the sum of the eigenvalues of the covariance Σ_p of the underlying distribution p. The sum of the eigenvalues captures the spread of a multivariate distribution with a single variable, similarly to the variance in the univariate case. To exploit this uncertainty estimation for our problem, we now extend E_{pose} to model not only the current and past poses but also the future ones and condition it on the choice of the future drone position. To determine the best next drone pose, we sample candidate positions and chose the one with the lowest uncertainty. This process is illustrated in Figure 3.

Future pose forecasting. In our setting, accounting for the dynamic motion of the person is key to successfully positioning the camera. We model the motion of the person from the current frame t to the next M future frames $t+i$, $i \in (1, \dots, M)$ linearly, i.e. we aim to keep the velocity of the joints constant across our window of frames. We also constrain the future poses by the bone length term. The future pose vectors $\boldsymbol{\Theta}^{t+i}$ are constrained by the smoothness and bone length terms, but for now not by any image-based term since the future images are not yet available at time t. Minimizing this extended E_{pose} for future poses gives the MAP poses $\hat{\boldsymbol{\Theta}}^{t+i}$. It continues the motion $\hat{\boldsymbol{\Theta}}^{t-k,\cdots,t+K}$ smoothly while maintaining the bone lengths. As we predict only the near future, we have found this simple extrapolation to be sufficient. We leave as future work the use of more advanced methods [10, 42] to forecast further.

Future measurement forecasting. We aim to find the future drone position, $\mathbf{D}^{t+1}$, that reduces the posterior uncertainty, but we do not have footage from future viewpoints to condition the posterior on. Instead, we use the predicted future human pose $\hat{\boldsymbol{\Theta}}^{t+i}$, $i \in (1, \dots, M)$, as a proxy for $\mathbf{L}^{t+i}$ and approximate $\mathbf{M}^{t+i}$ with the projection

$$\hat{\mathbf{M}}^{t+1} = \Pi(\hat{\mathbf{\Theta}}^{t+1}, \mathbf{D}^{t+1}, \boldsymbol{K}) \ . \tag{8}$$

At first glance, constraining the future pose on these virtual estimates in E_{pose} does not add anything since the terms E_{proj} and E_{lift} are zero at $\hat{\mathbf{\Theta}}^{t+1}$ by this construction. However, it changes the energy landscape and models how strong a future observation would constrain the pose posterior. In particular, the projection term, E_{proj}, narrows down the solution space in the direction of the image plane but cannot constrain it in the depth direction, creating an elliptical uncertainty as visualized in Fig 3. The combined influence of all terms is conveniently modeled as the energy landscape of E_{pose} and its corresponding posterior.

In our current implementation we assume that the 2D and 3D detections are affected by pose-independent noise, and their variance is captured by ω_p and ω_l, respectively. These factors could, in principle, be view dependent and in relation to the person's pose. For instance, [4] may be more accurate at reconstructing a front view than a side view. However, while estimating the uncertainty in deep networks is an active research field [26], predicting the expected uncertainty for an unobserved view has not yet been attempted for pose estimation. It is an interesting future work direction.

Variance estimator. E_{pose} and its corresponding posterior has a complex form due to the projection and prior terms. Hence, the sought-after covariance Σ_p cannot be expressed in closed form and approximating it by sampling the space of all possible poses would be expensive. Instead, for the sake of uncertainty estimation, we approximate $p(\mathbf{\Theta}|\mathbf{D}, \mathbf{M}, \mathbf{L}, \mathbf{b})$ locally with a Gaussian distribution q, such that

$$\Sigma_{p(\mathbf{\Theta}|\mathbf{D},\mathbf{M},\mathbf{L})} \approx \Sigma_q \text{ where } q = N(\mathbf{\Theta}|\hat{\mathbf{\Theta}}, \Sigma_q) \ , \tag{9}$$

with $\hat{\mathbf{\Theta}}$ and Σ_q the Gaussians mean and covariance matrix, respectively. Such an approximation is exemplified in Figure 2. For a Gaussian, the covariance of q can be computed in closed form as the inverse of the Hessian of the negative log likelihood, $\Sigma_q = H^{-1}_{-\log q}$, where $H_{-\log q} = \frac{\partial^2 - \log q(\mathbf{\Theta})}{\partial \mathbf{\Theta}}\Big|_{\mathbf{\Theta}=\hat{\mathbf{\Theta}}}$. Under the Gaussian assumption, Σ_p is thereby well approximated by the second order gradients, $H^{-1}_{E_{\text{pose}}}$, of E_{pose}. Our experiments show that this simplification holds well for the introduced error terms.

To select the view with minimum uncertainty among a set of K candidate drone trajectories, we therefore

1. optimize E_{pose} once to forecast M human poses $\hat{\mathbf{\Theta}}^{t+i}$, for $1 \leq i \leq M$
2. use these forecasted poses to set $\hat{\mathbf{L}}^{t+i}$ and $\hat{\mathbf{M}}^{t+i}$ for each $1 \leq i \leq M$ for each candidate trajectory c,
3. compute the second order derivatives of E_{pose} for each c, which form H_c, and
4. compute and sum up the respective eigenvalues to select the candidate with the least uncertainty.

Discussion. In principle, $p(\mathbf{\Theta}|\mathbf{M}, \mathbf{D}, \mathbf{L}, \mathbf{b})$, i.e. the probability of the most likely pose, could also act as a measure of certainty, as implicitly used in [27] on a known motion trajectory to minimize triangulation error. However, the term $E_{\text{proj}}(\hat{\mathbf{\Theta}}, \hat{\mathbf{M}})$ of E_{pose} is zero for the future time step $t+i$, because the projection of $\hat{\mathbf{\Theta}}^{t+i}$ is by construction equal to $\hat{\mathbf{M}}^{t+i}$ and therefore uninformative. Another alternative that has been proposed in the literature is to approximate the covariance through first order estimates [37], as a function of the Jakobi matrix. However, as also the first order gradients of E_{proj} vanish at the MAP estimate, this approximation is not possible in our case.

3.3. Drone Control Policies and Flight Model

In the experiments where we simulate drone flight, the algorithm decides between 9 candidate trajectories in the directions up, down, left, right, up-right, up-left, down-right, down-left and center. To ensure that the drone stays a fixed distance away from the person, the direction vector is normalized by the fixed-distance value.

In the remainder of this section, we describe how we model the flight of the drone so that we can predict the position of the drone along a potential trajectory in future time steps. By forecasting the future M locations of the drone on a potential trajectory c, we can predict the 2D pose estimations $\hat{\mathbf{M}}^{t+i}$ for each $\{i\}_{i=1}^{M}$ more accurately.

We control the flight of our drone by passing it the desired velocity vector and the desired yaw rotation amount with the maximum speed kept constant at 5 m/s. The drone is sent new commands once every $\Delta t = 0.2$ seconds.

We model the drone flight in the following manner. We assume that the drone moves with constant acceleration during a time step Δt. If the drone has current position x_{current} and velocity V_{current}, then with an current acceleration a_{current}, its next position x_{goal} in Δt time will be

$$x_{\text{goal}} = x_{\text{current}} + V_{\text{current}}\Delta t + 0.5 a_{\text{current}}\Delta t^2 \ . \tag{10}$$

The current acceleration at time t is found as a weighted average of the input acceleration a_{input} and the acceleration of the previous step a_{previous}. This can be written as

$$a_{\text{current}} = \alpha a_{\text{input}} + (1-\alpha) a_{\text{previous}}. \tag{11}$$

a_{input} is determined according to the candidate trajectory being evaluated. The direction of the acceleration vector is set to the direction of the candidate trajectory. We determine the magnitude of the input acceleration through least-square minimization of the difference between the predicted x_{goal} and the actual drone position. α is found by line search.

By estimating the future positions of the drone, we are able to forecast more accurate future 2D pose estimations,

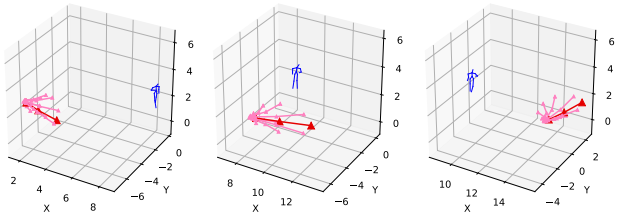

Figure 4. **Predicted trajectories** as the drone is circling the subject. The future drone positions are predicted for the future 3 steps, represented by triangle markers on the trajectories. Red depicts the chosen trajectory.

leading to more accurate decision making. Examples of predicted trajectories are shown in Figure 4. Further details are provided in the supplementary material.

4. Evaluation

In this section we evaluate the improvement on 3D human pose estimation that is achieved through optimization of the drone flight.

Simulation environment. Although [28, 3, 36] run in real time, and online SLAM from a monocular camera [9] is possible, we use a drone simulator since the integration of all components onto constrained drone hardware is difficult and beyond our expertise. We make simulation realistic by driving our characters with real motion capture data from the CMU Graphics Lab Motion Capture Database [1] and using the AirSim [33] drone simulator that builds upon the Unreal game engine and therefore produces realistic images of natural environments. Simulation also has the advantage that the same experiment can be repeated with different parameters and be directly compared to baseline methods and ground-truth motion.

Simulated test set. We test our approach on three CMU motions of increasing difficulty: *Walking* straight (subject 2, trial 1), *Dance* with twirling (subject 5, trial 8), and *Running* in a circle (subject 38, trial 3). Additionally, we use a validation set consisting of *Basketball* dribble (subject 6, trial 13), and *Sitting* on a stool (subject 13, trial 6), to conduct a grid search for hyperparameters.

Real test set. To show that our planner also works outside the simulator, we evaluate our approach on a section of the MPI-INF-3DHP dataset, which includes motions such as running around in a circle and waving arms in the air. The dataset provides 14 fixed viewpoints that are at varying distances from one another and from the subject, as depicted in Figure 6. In this case, the best next view is restricted to one of the 14 fixed viewpoints. This dataset lets us evaluate whether the object detector of [28], the 2D pose estimation method of [4], and the 3D pose regression technique of [36]

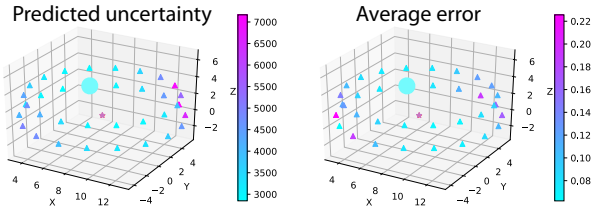

Figure 5. **Uncertainties estimates** across potential viewpoints (left image) compared with the average error we would obtain if we were to visit these locations (right image). The star represents the location of the subject and the large circle depicts the chosen viewpoint according to the lowest uncertainty.

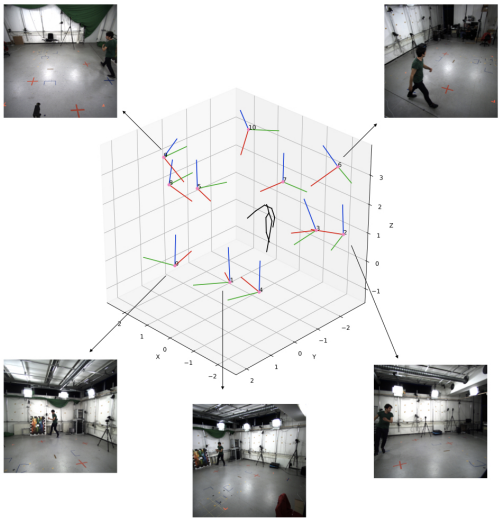

Figure 6. **MPI_INF_3DHP dataset**, which has images taken from 14 viewpoints with various distances to the subject. We use this dataset to evaluate our performance on datasets with realistic camera positioning and real images.

are reliable enough in real environments. Since we cannot control the camera in this setting, we remove those cameras from the candidate locations where we predict that the subject will be out of the viewpoint.

Baselines. Existing drone-based pose estimation methods use predefined policies to control the drone position relative to the human. Either the human is followed from a constant angle and the angle is set externally by the user [19] or the drone undergoes a constant rotation around the human [45]. As another baseline, we use a random decision policy, where the drone picks uniformly randomly among the proposed viewpoints. Finally, the oracle is obtained by moving the drone to the viewpoint where the reconstruction in the next time step will have the lowest average error, which is achieved by exhaustively trying all viewpoints *with* the corresponding image in the next time frame.

Hyper parameters. We set the weights of the loss term for the reconstruction as follows: $\omega_p = 0.0001$ (projec-

	Noisy ground truth				Networks	
	CMU-Walk	CMU-Dance	CMU-Run	MPI-INF-3DHP	MPI-INF-3DHP	Total
Oracle	0.101±0.001	0.101±0.001	0.109±0.001	0.136±0.002	0.17±0.0005	0.142±0.027
Ours (Active)	**0.113**±0.001	**0.116**±0.003	**0.19**±0.001	**0.145**±0.006	**0.21**±0.0008	**0.155**±0.39
Random	0.123±0.002	0.125±0.003	0.159±0.003	0.286±0.027	0.28±0.03	0.195±0.07
Constant Rotation	0.157±0.002	0.146±0.004	0.223±0.003	0.265±0.010	0.29±0.03	0.216±0.06
Constant Angle	0.895±0.54	0.683±0.31	0.985±0.24	1.73±0.61	1.26±0.53	1.11±0.36

Table 1. **3D pose accuracy on the teleportation experiment**, using noisy ground truth to estimate $\mathbf{M}$ and $\mathbf{L}$ in the first three columns, and using the networks of [43, 36] in the fourth column. We outperform all predefined baseline trajectories and approach the accuracy of the oracle that has access to the average error of each candidate position.

tion), $\omega_s = 1$ (smoothness), $\omega_l = 0.1$ (lift term), $\omega_b = 1$ (bone length), which were found by grid search. We set the weights for the decision making as $\omega_p = 0.001$, $\omega_s = 1$, $\omega_l = 0.1$, $\omega_b = 1$. Our reasoning is, we need to set the weights of the projection and lift terms slightly lower because they are estimated with large noise, which is introduced by the neural networks or as additive noise. However, they do not need to be as low for the uncertainty estimation.

4.1. Analyzing Reconstruction Accuracy

We report the mean Euclidean distance per joint in meters in the middle frame of the temporal window we optimize over. For teleportation mode, the size of the temporal window is set to $k = 2$ past frames and 1 future frame, and for the drone flight simulations, to $k = 6$ for past frames and 3 future frames.

Simulation Initialization. The frames are initialized by *back-projecting* the 2D joint locations estimated in the first frame, $\mathbf{M}^{t=0}$, to a distance d from the camera that is chosen such that the back-projected bone lengths match with the average human height. We then refine this initialization by running the optimization without the smoothness term, as there is only one frame. All the sequences are evaluated for 120 frames, with the animation sequences played at 5 Hz.

Teleportation Mode. To understand whether our uncertainty predictions for potential viewpoints coincide with the actual 3D pose errors we will have at these locations, we run the following simulation: We sample a total of 18 points on a ring around the person, as shown in Fig. 5, and allow the drone to teleport to any of these points. We optimize over a total of $k = 2$ past frames and forecast 1 frame into the future. We chose this window size to emphasize the importance of the next choice of frame.

We perform two variants of this experiment. In the first one, we simulate the 2D and 3D pose estimates, $\mathbf{M}, \mathbf{L}$, by adding Gaussian noise to the ground-truth data. The mean and standard deviation of this noise is set as the error of [3] and [36], run on the validation set of animations. Figure 7 shows a comparison between the ground truth values, noisy ground truth values and the network results. The results of this experiment are reported in Table 1, where we also provide the standard deviations across 5 trials with varying noise and starting from different viewpoints. On the MPI-INF-3DHP dataset, we also provide results using [3] and [36] on the simulator images to obtain the 2D and 3D pose estimates. Further results are in the supplementary material.

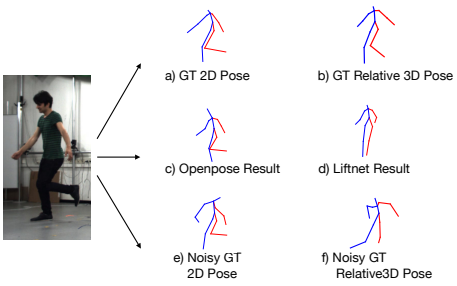

Figure 7. **Example image from the MPI-INF-3DHP dataset** along with the 2D pose detections $\mathbf{M}$ and 3D relative pose detections $\mathbf{L}$ obtained using ground truth, noisy ground truth or the networks of [3] and [36]. The noise we add on the ground truth poses is determined according to the statistics of [3] and [36], measured on our validation set.

Altogether, the results show that our active motion planner achieves consistently lower error values than the baselines and we come the closest to achieving the best possible error for these sequences and viewpoints, despite having no access to the true error. The random baseline also performs quite well in these experiments, as it takes advantage of the drone teleporting to a varied set of viewpoints. The trajectories generated by our active planner and the baselines is depicted in Figure 8. Importantly, Figure 5 evidences that our predicted uncertainties accurately reflect the true pose errors, thus making them well suited to our goal.

Simulating Drone Flight. To evaluate more realistic cases where the drone is actively controlled and constrained to only move to nearby locations, we simulate the drone flight using the AirSim environment. While simulating drone flight, we target a fixed radius of 7m from the subject and therefore provide direction candidates that lead to preserving this distance. We do not provide samples at different distances, as moving closer is unsafe and moving farther leads to more concentrated image projections and thus higher 3D errors. We also restrict the drone from flying outside the altitude range 0.25m-3.5m, so as to avoid crashing into the ground and flying above the subject.

In this set of experiments, we *fly* the drone using the

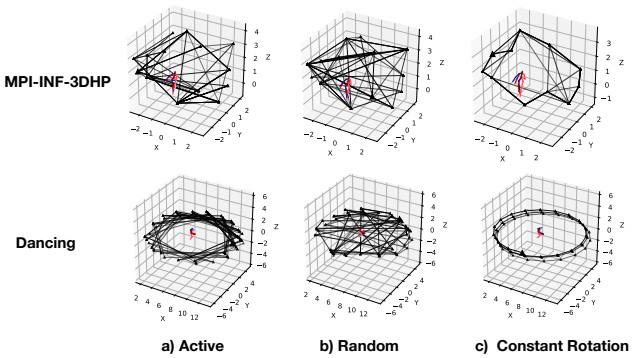

Figure 8. **Trajectories found by our active planner** along with random and constant rotation baselines. The first row depicts the trajectories for the MPI-INF-3DHP dataset, and the second row shows the trajectories for the dancing motion. The trajectories obtained with our algorithm are regular and look different from the random trajectories, especially for the dancing motion. Our algorithm prefers trajectories resulting in large angular variance with respect to the subject between viewpoints.

	CMU-Walk	CMU-Dance	CMU-Run	Total
Ours (Active)	**0.26**±0.03	0.22±0.04	0.44±0.04	0.31±0.10
Constant Rotation	0.28±0.06	**0.21**±0.04	**0.41**±0.02	**0.30**±0.08
Random	0.60±0.13	0.44±0.19	0.81±0.16	0.62±0.15
Constant Angle	0.41±0.07	0.63± 0.06	1.26±0.17	0.77±0.36

Table 2. **Results of drone full flight simulation**, using noisy ground truth as input to estimate $\mathbf{M}$ and $\mathbf{L}$. The results of constant rotation are the average of 10 runs, with 5 runs rotating clockwise and 5 counter-clockwise. Our approach yields results comparable to those of constant rotation, outperforming the other baselines. The trajectory our algorithm draws also results in a constant rotation, the only difference being the rotation direction.

simulator's realistic physics engine. To this end, we sample 9 candidate directions towards up, down, left, right, up-right, up-left, down-right, down-left and center. We then predict the 3 consecutive future locations using our simplified (closed form) physics model, to get and estimate where the drone will be at when continuing in each of the 9 directions. We then estimate the uncertainty at these sampled viewpoints and choose the minimum.

We achieve comparable results to constant rotation on simulated drone flight. In fact, except for the first few frames where the drone starts flying, we observe the same trajectory as constant rotation, only the rotation direction varies. Constant rotation being optimal in this setting is not counter-intuitive, as constant rotation is very useful for preserving momentum. This allows the drone to sample viewpoints as far apart from one another as possible, while keeping the subject in view. Figure 9 depicts the different baseline trajectories and the active trajectory.

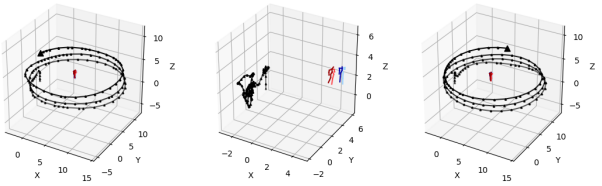

Figure 9. **Trajectories found during flight** by our active planner and the baselines. Our algorithm also chose to perform constant rotation. Because of the drone momentum, the random baseline cannot increase the distance between its camera viewpoints.

5. Conclusion and Future Work

We have proposed a theoretical framework for estimating the uncertainty of future measurements from a viewpoint. This permits us to improve 3D human pose estimation by optimizing the viewpoint selection to visit those locations with the lowest expected uncertainty. We have demonstrated with increasingly complex examples, in simulation with synthetic and real footage, that this theory translates to closed-loop drone control and improves pose estimation accuracy. We envision our approach being developed further for improving the performance of athletes and performance artists. It is important to preserve the subjects' privacy in such autonomous systems. We encourage researchers to be sensitive to this issue.

Key to the success of our approach is the integration of several sources of uncertainty. Our primary goal was to make uncertainty estimation tractable, but further improvements are needed to run it on an embedded drone system. The current implementation runs at 0.1Hz, but the optimization is implemented in Python using the convenient but slow automatic differentiation of PyTorch to obtain second derivatives. Furthermore, we have considered a physically plausible drone model but neglected physical obstacles and virtual no-go areas that would restrict the possible flight trajectories. In the case of complex scenes with dynamic obstacles, we expect our algorithm to outperform any simple, predefined policy. Currently, we assume a constant error for the 2D and 3D pose estimates. In future work, we will investigate how to derive situation-dependent noise models of deep neural networks. Furthermore, we plan to study new ways of estimating the uncertainty of the deployed deep learning methods and extend our work to optimize drone trajectories for different computer vision tasks.

6. Acknowledgements

This work was supported in part by the Swiss National Science Foundation and by a Microsoft Joint Research Project.

References

[1] CMU Graphics Lab Motion Capture Database. mocap.cs.cmu.edu.

[2] A. Aissaoui, A. Ouafi, P. Pudlo, C. Gillet, Z.-E. Baarir, and A. Taleb-Ahmed. Designing a Camera Placement Assistance System for Human Motion Capture Based on a Guided Genetic Algorithm. *Virtual reality*, 22(1):13–23, 2018.

[3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Conference on Computer Vision and Pattern Recognition*, pages 1302–1310, 2017.

[4] Y. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting Human Dynamics from Static Images. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[5] X. Chen and J. Davis. Camera Placement Considering Occlusion for Robust Motion Capture. *Computer Graphics Laboratory, Stanford University, Tech. Rep*, 2(2.2):2, 2000.

[6] W. Cheng, L. Xu, L. Han, Y. Guo, and L. Fang. ihuman3d: Intelligent human body 3d reconstruction using a single flying camera. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1733–1741. ACM, 2018.

[7] S. Choudhury, A. K. G., Ranade, and D. Dey. Learning to gather information via imitation. In *International Conference on Robotics and Automation*, 2017.

[8] J. Daudelin and M. Campbell. An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects. *IEEE Robotics and Automation Letters*, 2(3):1540–1547, 2017.

[9] A. J. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-Time Single Camera Slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[10] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent Network Models for Human Dynamics. In *International Conference on Computer Vision*, 2015.

[11] C. Gebhardt, S. Stevsic, and O. Hilliges. Optimizing for Aesthetically Pleasing Quadrotor Camera Motion. *ACM Transactions on Graphics*, 37(4):90:1–90:11, 2018.

[12] B. Hepp, D. Dey, S. Sinha, A. Kapoor, N. Joshi, and O. Hilliges. Learn-To-Score: Efficient 3D Scene Exploration by Predicting View Utility. In *European Conference on Computer Vision*, 2018.

[13] B. Hepp, M. Nießner, and O. Hilliges. Plan3D: Viewpoint and Trajectory Optimization for Aerial Multi-View Stereo Reconstruction. *ACM Transactions on Graphics*, 38(1):4, 2018.

[14] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *International Conference on Robotics and Automation*, 2016.

[15] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[16] J. Martinez, R. Hossain, J. Romero, and J. Little. A Simple Yet Effective Baseline for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2017.

[17] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. In *International Conference on 3D Vision*, 2017.

[18] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics*, 2017.

[19] T. Nägeli, S. Oberholzer, S. Plüss, J. Alonso-Mora, and O. Hilliges. Flycon: Real-time environment-independent multi-view human pose estimation with aerial vehicles. *ACM Transactions on Graphics*, 2018.

[20] E. Palazzolo and C. Stachniss. Information-driven autonomous exploration for a vision-based mav. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:59, 2017.

[21] G. Pavlakos, X. Zhou, K. Derpanis, G. Konstantinos, and K. Daniilidis. Coarse-To-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[22] G. Pavlakos, X. Zhou, K. D. G. Konstantinos, and D. Kostas. Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[23] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-Supervised Training. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[24] A. Pirinen, E. Gärtner, and C. Sminchisescu. Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction. In *Advances in Neural Information Processing Systems 32*, pages 3907–3917. 2019.

[25] A.-I. Popa, M. Zanfir, and C. Sminchisescu. Deep Multitask Architecture for Integrated 2D and 3D Human Sensing. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[26] S. Prokudin, P. Gehler, and S. Nowozin. Deep Directional Statistics: Pose Estimation with Uncertainty Quantification. In *European Conference on Computer Vision*, pages 534–551, 2018.

[27] P. Rahimian and J. K. Kearney. Optimal Camera Placement for Motion Capture Systems. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1209–1221, 2016.

[28] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. In *arXiv Preprint*, 2018.

[29] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: Egocentric Marker-Less Motion Capture with Two Fisheye Cameras. *ACM SIGGRAPH Asia*, 35(6), 2016.

[30] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, and N. Joshi. Submodular Trajectory Optimization for Aerial 3D Scanning. In *International Conference on Computer Vision*, 2017.

[31] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-Net: Localization-Classification-Regression for Human Pose. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[32] N. Saini, E. Price, R. Tallamraju, R. Enficiaud, R. Ludwig, I. Martinović, A. Ahmad, and M. Black. Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In *International Conference on Computer Vision*, Oct. 2019.

[33] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[34] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional Human Pose Regression. In *International Conference on Computer Vision*, 2017.

[35] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. Bülthoff, M. Black, and A. Ahmad. Active perception based formation control for multiple aerial vehicles. *IEEE Robotics and Automation Letters*, PP:1–1, 08 2019.

[36] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *International Conference on Computer Vision*, 2017.

[37] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, and A. Fitzgibbon. Online generative model personalization for hand tracking. *ACM Transactions on Graphics*, 36(6):243, 2017.

[38] D. Tome, C. Russell, and L. Agapito. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. In *arXiv Preprint*, 2017.

[39] D. Xiang, H. Joo, and Y. Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[40] L. Xu, L. Fang, W. Cheng, K. Guo, G. Zhou, Q. Dai, and Y. Liu. Flycap: Markerless motion capture using multiple autonomous flying cameras. *IEEE Transactions on Visualization and Computer Graphics*, PP, 10 2016.

[41] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes - the Importance of Multiple Scene Constraints. In *Conference on Computer Vision and Pattern Recognition*, June 2018.

[42] J. Y. Zhang, P. Felsen, A. Kanazawa, and J. Malik. Predicting 3d human dynamics from video. In *International Conference on Computer Vision*, 2019.

[43] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, and J. Feng. Multi-View Image Generation from a Single-View. In *arXiv Preprint*, 2017.

[44] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. We. Weakly-Supervised Transfer for 3D Human Pose Estimation in the Wild. In *arXiv Preprint*, 2017.

[45] X. Zhou, A. S. Liu, A. G. Pavlakos, A. V. Kumar, and K. Daniilidis. Human Motion Capture Using a Drone. In *International Conference on Robotics and Automation*, 2018.