Deep Reinforcement Learning for Active Human Pose Estimation

Erik Gärtner,[1*] **Aleksis Pirinen,**[1*] **Cristian Sminchisescu**[1,2,3]

[1]Department of Mathematics, Faculty of Engineering, Lund University
[2]Institute of Mathematics of the Romanian Academy
[3]Google Research
{erik.gartner, aleksis.pirinen, cristian.sminchisescu}@math.lth.se

Abstract

Most 3d human pose estimation methods assume that input – be it images of a scene collected from one or several viewpoints, or from a video – is given. Consequently, they focus on estimates leveraging prior knowledge and measurement by fusing information spatially and/or temporally, whenever available. In this paper we address the problem of an active observer with freedom to move and explore the scene spatially – in 'time-freeze' mode – and/or temporally, by selecting informative viewpoints that improve its estimation accuracy. Towards this end, we introduce *Pose-DRL*, a fully trainable deep reinforcement learning-based active pose estimation architecture which learns to select appropriate views, in space and time, to feed an underlying monocular pose estimator. We evaluate our model using single- and multi-target estimators with strong result in both settings. Our system further learns automatic stopping conditions in time and transition functions to the next temporal processing step in videos. In extensive experiments with the Panoptic multi-view setup, and for complex scenes containing multiple people, we show that our model learns to select viewpoints that yield significantly more accurate pose estimates compared to strong multi-view baselines.

1   Introduction

Existing human pose estimation models, be them designed for 2d or 3d reconstruction, assume that viewpoint selection is outside the control of the estimation agent. This problem is usually solved by a human, either once and for all, or by moving around and tracking the elements of interest in the scene. Consequently, the work is split between *sufficiency* (e.g. instrumenting the space with as many cameras as possible in motion capture setups), *minimalism* (work with as little as possible, ideally a single view, as given), or *pragmatism* (use whatever is available, e.g. a stereo system and lidar in a self-driving car). While each of these scenarios and their underlying methodologies make practical or conceptual sense in their context of applicability, none covers the case of an active observer moving in the scene in order to reduce uncertainty, with emphasis on trading accuracy and

computational complexity. There are good reasons for this, as experimenting with an active system faces the difficulty of linking perception and action in the real world, or may have to resort on simulation, which can however lack visual appearance and motion realism, especially for complex articulated and deformable structures such as people.

In this work we consider 3d human pose estimation from the perspective of an active observer, and operate with an idealization that allows us to distill the active vision concepts, develop new methodology, and test it on real image data. We work with a Panoptic massive camera grid (Joo et al. 2015), where we can both observe the scene in time-freeze, from a dense variety of viewpoints, and process the scene temporally, thus being able to emulate a moving observer. An active setup for 3d human pose estimation addresses the incomplete body pose observability in any monocular image due to depth ambiguities or occlusions (self-induced or produced by other people or objects). It also enables adaptation with respect to any potential limitations of the associated monocular pose estimation system, by sequentially selecting views that when combined yield accurate pose predictions.

In this context we introduce *Pose-DRL*, a deep reinforcement learning (RL) based active pose estimation architecture operating in a dense camera rig, which learns to select appropriate viewpoints to feed an underlying monocular pose predictor. Moreover, our model learns when to stop viewpoint exploration in time-freeze, or continue to the next temporal step when processing video. In evaluations using Panoptic we show that our system learns to select sets of views yielding more accurate pose estimates compared to strong multi-view baselines. The results not only show the advantage of intelligent viewpoint selection, but also that often 'less is more', as fusing too many possibly incorrect viewpoint estimates leads to inferior results.

As our model consists of a deep RL-based active vision module on top of a task module, it can be easily adjusted for other visual routines in the context of a multi-camera setup by simply replacing the task module and retraining the active vision component, or refining them jointly in case of access and compatibility. We show encouraging results using different pose estimators and task settings.

---

2 Related Work

Extracting 2d and 3d human representations from *given* images or video is a vast research area, recently fueled by progress in keypoint detection (Wei et al. 2016; Papandreou et al. 2017), semantic body parts segmentation (Popa, Zanfir, and Sminchisescu 2017), 3d human body models (Loper et al. 2015), and 3d motion capture data (Ionescu et al. 2014; von Marcard et al. 2018). Deep learning plays a key role in most human pose and shape estimation pipelines (Bogo et al. 2016; Rhodin et al. 2016; Popa, Zanfir, and Sminchisescu 2017; Pavlakos et al. 2017; Rogez, Weinzaepfel, and Schmid 2017; Zanfir, Marinoiu, and Sminchisescu 2018; Mehta et al. 2017; Kanazawa et al. 2018), sometimes in connection with non-linear refinement (Bogo et al. 2016; Zanfir, Marinoiu, and Sminchisescu 2018). Systems integrating detailed face, body and hand models have also been proposed (Joo, Simon, and Sheikh 2018). Even so, the monocular 3d case is challenging due to depth ambiguities which motivated the use of additional ordering constraints during training (Pavlakos, Zhou, and Daniilidis 2018).

In addition to recent literature for static pipelines, the community has recently seen an increased interest for active vision tasks, including RL-based visual navigation (Ammirato et al. 2017; Das et al. 2018; Xia et al. 2018; Zhu et al. 2017). In (Ammirato et al. 2017) a real-world dataset of sampled indoor locations along multiple viewing directions is introduced. An RL-agent is trained to navigate to views in which a given instance detector is accurate, similar in spirit to what we do, but in a different context and task.

A joint gripping and viewing policy is introduced in (Cheng, Agarwal, and Fragkiadaki 2018), also related to us in seeking policies that choose occlusion-free views. The authors of (Cheng, Wang, and Fragkiadaki 2018) introduce an active view selection system and jointly learn a geometry-aware model for constructing a 3d feature tensor, which is fused together from views predicted by a policy network. In contrast to us, their policy predicts one of 8 adjacent discrete camera locations, they do not consider moving objects, their model does not automatically stop view selection, and they do not use real data. In (Jayaraman and Grauman 2018; Xiong and Grauman 2018), active view selection is considered for panoramic completion and panorama projection, respectively. Differently from us, their view selection policies operate on discretized spheres and do not learn automatic stopping conditions. An approach for active multi-view object recognition is proposed in (Johns, Leutenegger, and Davison 2016), where pairs of images in a view trajectory are sequentially fed to a CNN for recognition and for next best view prediction. Optimization is done over discretized movements and pre-set trajectory lengths, in contrast to us.

Most related to us is (Pirinen, Gärtner, and Sminchisescu 2019), who also consider active view selection in the context of human pose estimation. However, they work with 2d joint detectors and learn to actively triangulate those into 3d pose reconstructions. Thus we face different challenges – while (Pirinen, Gärtner, and Sminchisescu 2019) only require each joint to be visible in two views for triangulation, our model has to consider which views yield accurate fused estimates. Furthermore, their model does not learn a stopping action

that trades accuracy for speed, and they do not study both the single-target and multi-target cases, as we do in this paper.

Aside from active vision applications in real or simulated environments, reinforcement learning has also been successfully applied to other vision tasks, e.g. object detection (Caicedo and Lazebnik 2015; Pirinen and Sminchisescu 2018), object tracking (Zhang et al. 2017; Yun et al. 2018) and visual question answering (Das et al. 2017).

3 Active Human Pose Estimation

In this section we describe our active human pose estimation framework, arguing it is a good proxy for a set of problems where an agent has to actively explore to understand the scene and integrate task relevant information. For example, a single view may only contain parts of the human body (or be absent of the person altogether) and the agent needs to find a better view to capture the person's pose. Pose estimators are often trained on a limited set of viewing angles and yield lower performance for others. Our setup forces the agent to also take any estimation limitations into account when selecting multiple views. In particular, we show in §5.1 that learning to find good views and fusing them is more important than relying on a large number of random ones, or the full available set, as standard – see also Fig. 4.

Concepts in the following sections will, for simplicity, be described assuming the model is estimating the pose of a single *target person* (though scenes may contain multiple people occluding the target). The setting in which *all* people are reconstructed simultaneously is described in §4.4.

3.1 Active Pose Estimation Setup

We idealize our active pose estimation setup using CMU's Panoptic installation (Joo et al. 2015) as it captures real video data of scenes with multiple people and cameras densely covering the viewing sphere. This allows us to simulate an active agent observing the scene from multiple views, without the complexity of actually moving a camera. It also enables controllable and reproducible experiments. The videos are captured in a large spherical dome fitted with synchronized HD cameras.[1] Inside the dome several human actors perform a range of movements, with 2d and 3d joint annotations available. The dataset is divided into a number of *scenes*, video recordings from all synchronized cameras capturing different actors and types of movements, ranging from simple pose demonstrations to intricate social games.

Terminology. We call a *time-freeze* $\{v_1^t, \ldots, v_N^t\}$ the collection of views from all N time-synchronized cameras at time t, with v_i^t the image (referred to as *view* or *viewpoint*) taken by camera i at time t. A subset of a time-freeze is an *active-view* $\mathcal{V}^t = \{v_1^t, \ldots, v_k^t\}$ containing k selected views from the time-freeze. A temporally contiguous sequence of active-views is referred to as an *active-sequence*, $\mathcal{S}^{1:T} = \{\mathcal{V}^1, \mathcal{V}^2, \ldots, \mathcal{V}^T\}$. We will often omit the time superfix t unless the context is unclear; most concepts

---

[1]There are about 30 cameras per scene. The HD cameras provide better image quality than VGA and are sufficiently dense, yet spread apart far enough to make each viewpoint unique.
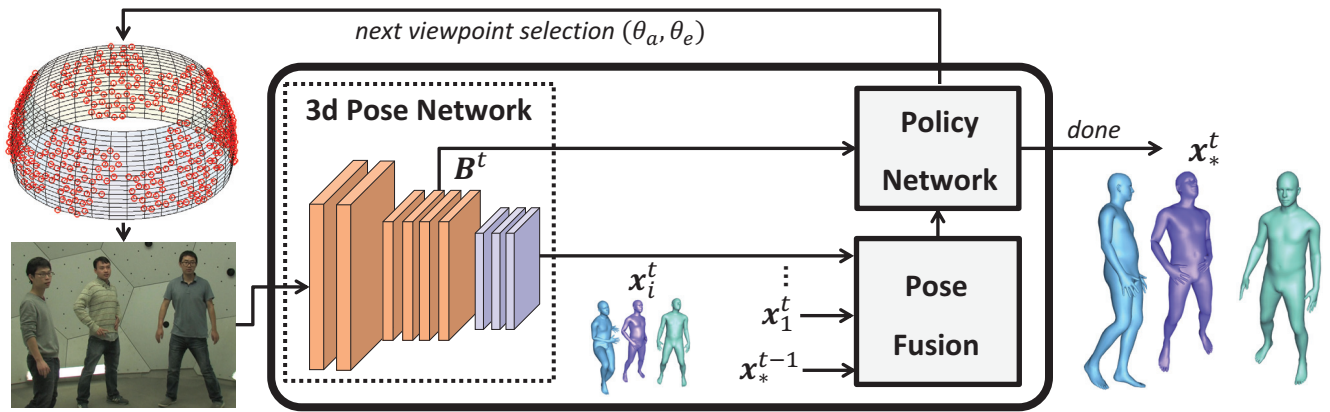
Figure 1: Overview of our Pose-DRL agent for active human pose estimation. The agent initially observes the scene from a randomly given camera on the rig. In each visited viewpoint, the associated image is processed by a 3d pose estimation network, producing the base state $\boldsymbol{B}^t$ of the agent and pose estimate(s) $\boldsymbol{x}_i^t$. The pose estimate is fused together with estimates from previous viewpoints $\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_{i-1}^t$ and the previous temporal step $\boldsymbol{x}_\star^{t-1}$. Both the current and fused estimate are fed as additional features to the agent. At each step the policy network outputs the next viewpoint, until it decides it is done and continues to next active-view at time $t+1$. The viewpoint selection action predicts spherical angles relative to the agent's current location on the camera rig, and the closest camera associated with the predicted angles is visited next. When the agent is done it outputs $\boldsymbol{x}_\star^t$, the per-joint fusion of the individual pose estimates seen during the current active-view and the fused pose estimate from the previous active-view, cf. (2). Pose-DRL can be used either to reconstruct a target person, or to reconstruct all people in the scene. The underlying pose estimator is exchangeable – we show strong results using two different ones in §5.1.

will be explained at the level of time-freezes. The image corresponding to a view v_i can be fed to a pose predictor to produce a pose estimate $\boldsymbol{x}_i \in \mathbb{R}^{45}$ ($15\times$ 3d joints).

Task definition. We define the task of *active pose estimation* at each time step as selecting views from a time-freeze to generate an active-view. The objective is to produce an accurate fused estimate $\boldsymbol{x}_\star$ from pose predictions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ associated with the active-view (k may vary between active-views). The deep pose estimation network is computationally demanding and therefore working with non-maximal sets of views decreases processing time. Moreover, it improves estimates by ignoring obstructed views, or those a given pose predictor cannot accurately handle. The goal of the full active pose estimation task is to produce accurate fused pose estimates over the full sequence, i.e., to produce an active-sequence with accurate corresponding fused pose estimates.

3.2 Detection and Matching of Multiple People

To solve active human pose estimation the model must address the problems of detecting, tracking, and distinguishing people in a scene. It must also be robust to variations in appearance since people are observed over time and from different views. We use Faster R-CNN (Ren et al. 2015) for detecting people. At the start of an active-sequence the agent is given appearance models, consisting of instance-sensitive features for each person. For each visited view, the agent computes instance features for all detected persons, comparing them with the given appearance models to identify the different people.

Obtaining appearance models. A generic instance classifier, implemented as a VGG-19 based (Simonyan and Zisserman 2015) siamese network, is trained for 40k iterations on the training set with a contrastive loss to distinguish between different persons. Each mini-batch (of size 16) consists of randomly sampled pairs of ground-truth crops of people in the training set. We ensure that the training is balanced by sampling pairs of person crops such that the probability of the two crops containing the same person is the same as that of containing two different persons. The people crops are sampled uniformly across scenes, spatially and temporally, yielding a robust instance classifier.

Once the instance classifier has been trained, we fine-tune it for 2k iterations for each scene and then use it to construct appearance models at the beginning of an active-sequence. For each person, we sample L instance features from time-freezes from the same scene, but outside of the time span of the current active-sequence to limit overfitting. Denote by $\boldsymbol{u}_i^l$, the i:th instance feature for the l:th person, with $i = 1, \ldots, L$. Then we set as appearance model:

$$\boldsymbol{m}^l = \mathrm{median}(\boldsymbol{u}_1^l, \ldots, \boldsymbol{u}_L^l) \qquad (1)$$

We set $L = 10$ to obtain a diverse set of instance features for each person, yielding a robust appearance model.

Stable matching of detections. In each visited viewpoint during an active-sequence, the agent computes instance features for all detected persons, comparing them with the given appearance models to identify the different people. To ensure a stable matching, we use the Hungarian algorithm. Specifically, the cost $c^{j,l}$ of matching the j:th detection with instance feature $\boldsymbol{u}^j$ in the current
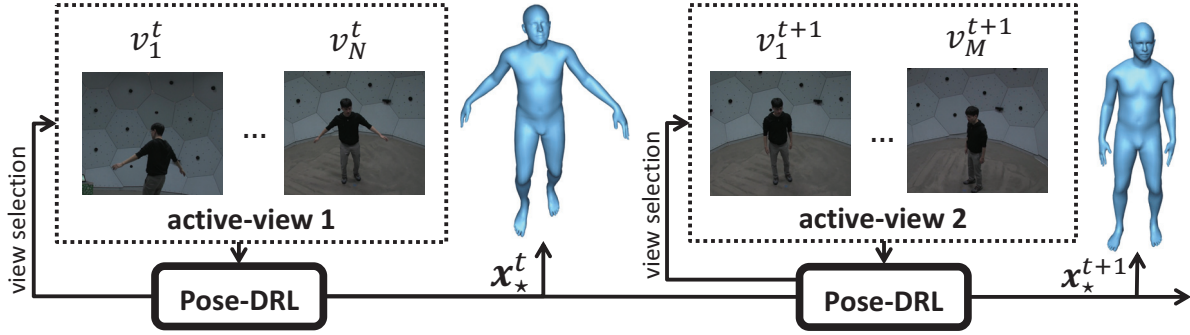
Figure 2: Illustration of how Pose-DRL operates on an active-sequence, here shown for a single-person scenario. Fused pose estimates are fed to subsequent active-views within the active-sequence, both as additional state representation for action selection, and for fusing poses temporally.

viewpoint to the appearance model m^l of the l:th person is $c^{j,l} = \|u^j - m^l\|_2^2$. Since the target person may not be visible in all viewpoints throughout the active-sequence, we specify a *cost threshold*, $\mathcal{C} = 0.5$, such that if the assignment cost $c^{j,l}$ of the target is above it (i.e. $c^{j,l} > \mathcal{C}$), we consider the person to not be visible in the view. In that case the associated pose is not fused into the final estimate.

4 Deep Reinforcement Learning Model

We now introduce our Pose-DRL agent for solving the active human pose estimation task and first explain the agent's state representation and actions, then define the reward signal for training an agent which selects views that yield an accurate fused pose estimate while keeping down processing time.

4.1 Overview of the Pose-DRL Agent

The agent is initiated at a randomly selected view v_1^1 in the first active-view $\mathcal{V}^1$ of an active-sequence $\mathcal{S}^{1:T}$. Within the current active-view $\mathcal{V}^t$, the agent issues *viewpoint selection* actions to progressively select a sequence of views $v_2^t, \ldots, v_k^t$, the number of which may vary between active-views. At each view v_i^t the underlying pose estimator predicts the pose x_i^t. As seen in Fig. 1 the cameras are approximately located on a partial sphere, so a viewpoint can be specified by the azimuth and elevation angles (referred to as *spherical angles*). Thus for viewpoint selection the Pose-DRL agent predicts spherical angles relative to its current location and selects the camera closest to those angles.

Once the agent is done exploring viewpoints associated to a particular time-freeze it issues the *continue* action and switches to the next active-view $\mathcal{V}^{t+1}$, at which time the collection of individual pose estimates x_i^t from the different viewpoints are fused together with the estimate $x_\star^{t-1}$ from the previous active-view $\mathcal{V}^{t-1}$:

$$x_\star^t = f(x_\star^{t-1}, x_1^t, x_2^t, \ldots, x_k^t) \qquad (2)$$

Including the previous time step estimate $x_\star^{t-1}$ in the pose fusion as in (2) often improves results (see §5.2). After returning the fused pose estimate $x_\star^t$ for the current active-view, the agent continues to the next active-view $\mathcal{V}^{t+1}$. The initial view v_1^{t+1} for $\mathcal{V}^{t+1}$ is set to the final view v_k^t of $\mathcal{V}_t$,

i.e., $v_1^{t+1} = v_k^t$. The process repeats until the end of the active-sequence. Fig. 1 and 2 show model overviews for active-views and active-sequences, respectively.

4.2 State-Action Representation

To simplify notation, we here describe how the agent operates in a given time-freeze, and in this context will use t to index actions within the active-view, as opposed to temporal structures. The state at step t is the tuple $S^t = (B^t, X^t, C^t, u^t)$. Here $B^t \in \mathbb{R}^{H \times W \times C}$ is a deep feature map associated with the underlying 3d pose estimation architecture. $X^t = \{x_t, \tilde{x}, x_\star^{hist}\}$ where x_t is the current individual pose estimate, $\tilde{x} = f(x_1, \ldots, x_t)$ is the current partially fused pose estimate, and $x_\star^{hist}$ is a history of fused predictions from 4 previous active-views. The matrix $C^t \in \mathbb{N}^{w \times h \times 2}$ consists of an *angle canvas*, a discretized encoding[2] of the previously visited regions on the camera rig, as well as a similar encoding of the camera distribution over the rig. Finally, $u^t \in \mathbb{R}^2$ is an auxiliary vector holding the number of actions taken and the number of people detected.

For action selection we use a deep stochastic policy $\pi_w(A^t|S^t)$ parametrized by w which predicts the action $A^t = \{\theta_a^t, \theta_e^t, c^t\}$. Here (θ_a^t, θ_e^t) is the azimuth-elevation angle pair, jointly referred to as *viewpoint selection*, and c^t is a Bernoulli variable indicating whether to continue to the next active-view (occurring if $c^t = 1$), referred to as the *continue* action. To produce action probabilities the base feature map B^t is fed through two convolutional blocks which are shared between the viewpoint selection and continue actions. The output of the second convolutional block is then concatenated with X^t, C^t and u^t and fed to viewpoint selection- and continue-branches with individual parameters. Both action branches consist of 3 fully-connected layers with $\tanh$ activations. The probability of issuing the continue action is computed using a sigmoid layer:

$$\pi_w(c^t = 1|S^t) = \sigma\left[w_c^\top z_c^t + b_c\right] \qquad (3)$$

---

[2]The camera sphere is discretized into w bins in the azimuth direction and h bins in the elevation direction, appropriately wrapped to account for periodicity. We set $w = 9$ and $h = 5$.

Model	# Views	Maf	Ult	Pose	Maf + Ult	All
Pose-DRL-S	auto	130.3 (4.6)	135.4 (3.4)	135.3 (3.7)	134.2 (3.8)	135.0 (3.7)
Pose-DRL-S	fixed	144.7 (5.0)	157.5 (4.0)	135.1 (4.0)	155.5 (4.0)	140.4 (4.0)
Rand-S	fixed	160.2 (5.0)	178.3 (4.0)	145.7 (4.0)	175.6 (4.0)	157.1 (4.0)
Max-Azim-S	fixed	156.3 (5.0)	171.4 (4.0)	139.9 (4.0)	169.4 (4.0)	150.3 (4.0)
Oracle-S	fixed	103.4 (5.0)	108.9 (4.0)	106.5 (4.0)	108.5 (4.0)	105.4 (4.0)

Model	# Views	Maf	Ult	Pose	Maf + Ult	All
Pose-DRL-M	auto	114.8 (7.5)	116.4 (6.6)	104.6 (2.1)	115.9 (6.8)	110.7 (4.5)
Pose-DRL-M	fixed	114.8 (8.0)	118.0 (7.0)	106.7 (3.0)	117.6 (7.0)	112.8 (5.0)
Rand-M	fixed	128.8 (8.0)	134.9 (7.0)	115.9 (3.0)	131.4 (7.0)	126.0 (5.0)
Max-Azim-M	fixed	123.5 (8.0)	131.2 (7.0)	116.3 (3.0)	131.6 (7.0)	126.4 (5.0)
Oracle-M	fixed	98.6 (8.0)	102.4 (7.0)	90.2 (3.0)	101.6 (7.0)	92.6 (5.0)

Table 1: Reconstruction error (mm/joint) for Pose-DRL and baselines on active-sequences on the selected Panoptic test splits. Results are shown both for the setting where the agent decides the number of views (auto) *and* when using a fixed number of views. In the latter case, the number of views is set to the closest integer corresponding to the average in auto mode, rounded up. The baselines are also evaluated at this preset number of views. The average number of views are shown in parentheses. Pose-DRL models which automatically select the number of views outperform the heuristic baselines and fixed Pose-DRL models on all data splits, despite fusing estimates from fewer views on average. Left: Single-target mode (S), using DMHS as pose estimator. The agent significantly outperforms the baselines (e.g. 35 mm/joint improvement over *Max-Azim* on multi-people data *Maf + Ult*). Right: Multi-target mode (M), using MubyNet as pose estimator. MubyNet is a more recent and accurate estimator, so the average errors are typically lower than the DMHS-counterparts. Automatic termination is useful in the multi-target setting as well, although it does not provide as drastic gains as in the single-target setup.

where w_c and b_c are trainable weights and bias, and z_c^t is the output from the penultimate fully-connected layer of the *continue* action branch.

Due to the periodic nature of the viewpoint prediction task we rely on von Mises distributions for sampling the spherical angles. We use individual distributions for the azimuth and elevation angles. The probability density function for the azimuth is given by:

$$\pi_{\boldsymbol{w}}\left(\theta_a^t | S^t\right) = \frac{1}{2\pi I_0(m_a)} \exp\{m_a \cos(\theta_a^t - \tilde{\theta}_a(\boldsymbol{w}_a^\top \boldsymbol{z}_a^t + b_a))\} \quad (4)$$

where I_0 is the zeroth-order Bessel function, normalizing (4) to a proper probability distribution over the unit circle $[-\pi, \pi]$. Here $\tilde{\theta}^a$ is the mean of the distribution (parametrized by the neural network), m_a is the precision parameter,[3] $\boldsymbol{w}_a$ and b_a are trainable weights and bias, respectively, and $\boldsymbol{z}_a^t$ comes from the penultimate fully-connected layer of the viewpoint selection action branch. The support for the azimuth angle should be on a full circle $[-\pi, \pi]$, and hence we set

$$\tilde{\theta}_a(\boldsymbol{w}_a^\top \boldsymbol{z}_a^t + b_a) = \pi \tanh(\boldsymbol{w}_a^\top \boldsymbol{z}_a^t + b_a) \quad (5)$$

The probability density function for the elevation angle has the same form (4) as that for the azimuth. However, as seen in Fig. 1, the range of elevation angles is more limited than for the azimuth angles. We denote this range $[-\kappa, \kappa]$ and the mean elevation angle thus becomes[4]

$$\tilde{\theta}_e(\boldsymbol{w}_e^\top \boldsymbol{z}_e^t + b_e) = \kappa \tanh(\boldsymbol{w}_e^\top \boldsymbol{z}_e^t + b_e) \quad (6)$$

In practice, when sampling elevation angles from the von Mises, we reject samples outside the range $[-\kappa, \kappa]$.

---

[3] We treat the precision parameters as constants but increase them over training to focus the policy on high-reward viewpoints.

[4] With notation analogous to that of the azimuth angle, cf. (5).

4.3 Reward Signal for Policy Gradient Objective

The agent should strike a balance between choosing sufficiently many cameras so the resulting 3d pose estimate is as accurate as possible, while ensuring that not too many cameras are visited, to save processing time. As described earlier, the two types of actions are *viewpoint selection* and *continue*. We will next cover the reward functions for them.

Viewpoint selection reward. At the end of an active-view we give a reward which is inversely proportional to the ratio between the final and initial reconstruction errors within the active-view. We also give a penalty $\epsilon = 2.5$ each time the agent goes to an already visited viewpoint. Thus the viewpoint selection reward is:

$$r_v^t = \begin{cases} 0, & \text{if } c^t = 0 \text{ and view not visited} \\ -\epsilon, & \text{if } c^t = 0 \text{ and view visited before} \\ 1 - \frac{\varepsilon^k}{\varepsilon^1}, & \text{if } c^t = 1 \end{cases} \quad (7)$$

where k is the number of views visited prior to the agent issuing the *continue* action ($c^t = 1$), ε^1 is the reconstruction error associated with the initial viewpoint, and ε^k denotes the final reconstruction error, i.e. $\varepsilon^k = \|\boldsymbol{x}_\star - \boldsymbol{x}_{\text{gt}}\|_2^2$. Here $\boldsymbol{x}_\star$ is the final fused pose estimate for the active-view, cf. (2), and $\boldsymbol{x}_{\text{gt}}$ is the ground-truth 3d pose for the time-freeze.

Continue action reward. The *continue* action has two purposes: *(i)* ensure that not too many viewpoints are visited to reduce computation time, and *(ii)* stop before suboptimal viewpoints are explored, which could happen if the agent is forced to visit a preset number of viewpoints. Therefore, the *continue* action reward is:

$$r_c^t = \begin{cases} 1 - \frac{\min\limits_{j \in \{t+1, \ldots, k\}} \varepsilon^j}{\varepsilon^t} - \tau, & \text{if } c^t = 0 \\ 1 - \frac{\varepsilon^k}{\varepsilon^1}, & \text{if } c^t = 1 \end{cases} \quad (8)$$

At each step that the agent decides *not* to continue to the next active-view ($c^t = 0$), the agent is rewarded relative to
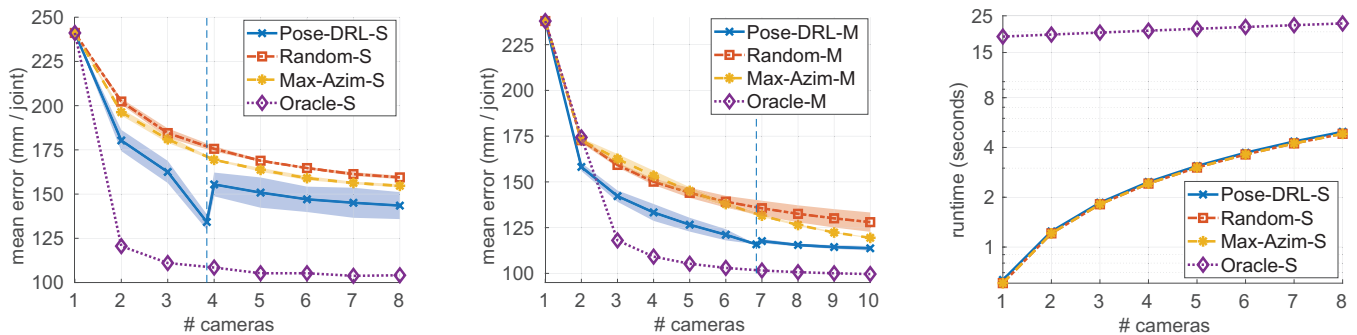
Figure 3: How the number of views affects pose estimation error and runtimes of Pose-DRL and baselines on multi-people data (union of *Mafia* and *Ultimatum* test sets). We show mean and 95% confidence intervals over 5 seeds. Left: Reconstructing a single target person. Estimation error reduces with added viewpoints, and the agent consistently outperforms the non-oracle baselines. The automatic *continue* action (dashed line at 3.8 views on average) yields significantly lower reconstruction errors than any fixed viewpoint schemes. Hence the auto-model clearly provides the best speed-accuracy trade-off. Middle: Simultaneously reconstructing all persons. The agent outperforms the heuristic baselines in this setting too. Adaptively determining when to continue to the next active-view (6.8 views on average) yields better results than fusing from 7 cameras all the time. The gain is not as pronounced as in the single-target case, since more viewpoints typically leads to increased estimation accuracy for some of the persons. Right: Runtime of the Pose-DRL agent and baselines vs. number of views (log scale). The oracle always needs to evaluate the deep pose estimation system and detector for all cameras due to its need to sort from best to worst, independently of the number of viewpoints, which explains its high runtime. Our agent is as fast as the heuristic baselines.

the ratio between the error at the best future stopping point within the active-view (with lowest reconstruction error) and the error at the current step. If in the future the agent selects viewpoints that yield lower reconstruction error the agent is rewarded, and vice verse if the best future error is higher. In addition, the agent gets a penalty τ at each step, which acts as an *improvement threshold*, causing the reward to become negative unless the ratio is above the specified threshold τ. This encourages the agent not to visit many viewpoints in the current active-view unless the improvement is above the given threshold. On the validation set we found $\tau = 0.07$ to provide a good balance.

Policy gradient objective. We train the Pose-DRL network in a policy gradient framework, maximizing expected cumulative reward on the training set with objective

$$J(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{s} \sim \pi_{\boldsymbol{w}}} \left[\sum_{t=1}^{|\boldsymbol{s}|} r^t \right] \tag{9}$$

where $\boldsymbol{s}$ denotes state-action trajectories, and the reward signal $r^t = r_v^t + r_c^t$, cf. (7) - (8). We approximate the gradient of the objective (9) using REINFORCE (Williams 1992).

4.4 Active Pose Estimation of Multiple People

So far we have explained the Pose-DRL system that estimates the pose of a target person, assuming it is equipped with a detection-based single person estimator. This system can in principle estimate multiple people by generating active-sequences for each person individually. However, to find a *single* active-sequence that reconstructs *all* persons, one can equip Pose-DRL with an image-level multi-people estimator instead. In that case, the state representation is modified to use the image level feature blob from

the multi-people estimator ($\boldsymbol{B}_t$ in Fig. 1). The reward signal used when learning to reconstruct all people is identical to (7) - (8), except that the rewards are averaged over the individual pose estimates. Thus Pose-DRL is very adaptable in that the underlying pose estimator can easily be changed.

5 Experiments

Dataset. We use diverse scenes for demonstrating and comparing our active pose estimation system, considering complex scenes with multiple people (*Mafia*, *Ultimatum*) as well as single person ones (*Pose*). The motions range from basic poses to various social games. Panoptic provides data as 30 FPS-videos which we sample to 2 FPS, making the data more manageable in size. It also increases the change in pose between consecutive frames.

The data we use consists of the same 20 scenes as in (Pirinen, Gärtner, and Sminchisescu 2019). The scenes are randomly split into training, validation and test sets with 10, 4 and 6 scenes, respectively. Since we split the data over the scenes, the agent needs to learn a general look-around-policy which adapts to various circumstances (scenarios and people differ between scenes). All model selection is performed exclusively on the training and validation sets; final evaluations are performed on the test set. The data consists of 343k images, of which 140k are single-person and 203k are multi-people scenes.

Implementation details. We attach Pose-DRL on top of the DMHS monocular pose estimation system (Popa, Zanfir, and Sminchisescu 2017). In the multi-people setting described in §4.4 we instead use MubyNet (Zanfir et al. 2018). Both estimators were trained on Human3.6M (Ionescu et al. 2014). To avoid overfitting we do not to fine-tune these on Panoptic, and instead emphasize how

Pose-DRL can select good views with respect to the underlying estimation system (but joint training is possible). We use an *identical set of hyperparameters* when using DMHS and MubyNet, except the improvement threshold τ, which is -0.07 for DMHS and -0.04 for MubyNet, which shows that Pose-DRL is robust with respect to the pose estimator used. We use median averaging for fusing poses, cf. (2).

Training. We use 5 active-sequences, each consisting of length 10, to approximate the policy gradient, and update the policy parameters using Adam (Kingma and Ba 2015). As standard, to reduce variance we normalize cumulative rewards for each episode to zero mean and unit variance over the batch. The maximum trajectory length is set to 8 views including the initial one (10 in the multi-target mode, as it may require more views to reconstruct all people). The *viewpoint selection* and *continue* actions are trained jointly for 80k episodes. The learning rate is initially set to `5e-7` and is halved at 720k and 1440k agent steps. We linearly increase the precision parameters m_a and m_e of the von Mises distributions from $(1, 10)$ to $(25, 50)$ in training, making the viewpoint selection increasingly focused on high-rewarding regions as training proceeds.

Baselines. To evaluate our active human pose estimation system we compare it to several baselines, similar to (Pirinen, Gärtner, and Sminchisescu 2019). For fair comparisons, the baselines use the same pose estimator, detector and matching approach. All methods obtain the same initial random view as the agent at the start of the active-sequence. We design the following baselines: i) *Random:* Selects k different random views; ii) *Max-Azim:* Selects k different views equidistantly with respect to the azimuth angle. At each azimuth angle it selects a random elevation angle; iii) *Oracle:* Selects as next viewpoint the one that minimizes the fused 3d pose reconstruction when combined with pose estimates from all viewpoints observed so far (averaged over all people in the multi-target setting). This baseline cheats by extensively using ground-truth information, and thus it shown as a lower bound with respect to reconstruction error. In addition to cheating during viewpoint selection, the oracle is also impractically slow since it requires computing pose estimates for *all* available viewpoints and exhaustively computing errors for all cameras in each step.

5.1 Quantitative Results

We report results both for the Pose-DRL agent that tracks and reconstructs a single target person (possibly in crowded scenes) and for the Pose-DRL model which actively estimates poses for all persons in the scene, cf. §4.4. Pose-DRL is trained over 5 different random initializations of the policy network, and we report average results. In each case, training the model 80k steps gave best results on the validation set, so we use that. Also, for the heuristic baselines we report average results over 5 seeds (the oracle is deterministic).

Our agent is compared to the baselines on the Panoptic test set on active-sequences consisting of 10 active-views. Table 1 presents reconstruction errors. Fig. 3 shows how the the number of selected views affects accuracy and runtimes.
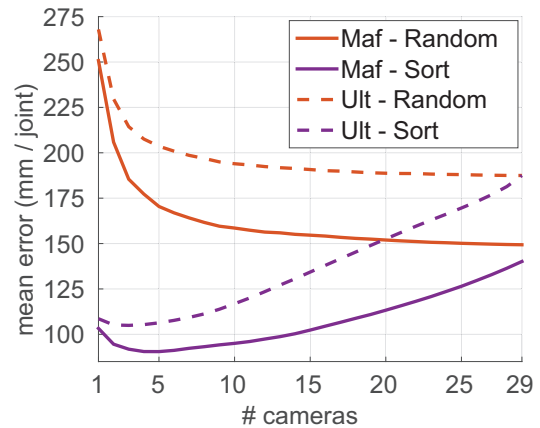

Figure 4: Per-joint pose reconstruction error for the monocular human pose estimation architecture DMHS vs. number of viewpoints, both when randomly choosing viewpoints, and when using an sorting strategy which selects viewpoints in ascending order of individual reconstruction error (note that this requires ground-truth). Results shown for multi-people data (*Mafia*, *Ultimatum*) on the CMU Panoptic dataset. For a good viewpoint selection policy such as *Sort*, estimation accuracy only improves when adding a few extra cameras, but then begins to deteriorate, indicating the need to adaptively terminate viewpoint selection early enough.

For visualizations[5] of Pose-DRL, see Fig. 5 - 7.

Single-target estimation. It is clear from Table 1 (left) and Fig. 3 (left) that Pose-DRL outperforms the heuristic baselines, which is particularly pronounced for multi-people data. In such scenes, the view selection process is more delicate, as it requires avoiding cameras where the target is occluded. We note that the automatically stopping agent yields by far the most accurate estimates, which shows that it is capable of continuing to the next active-view when it is likely that the current one does not provide any more good views. Thus it is often better to fuse a few accurate estimates than including a larger set of poorer ones.

Multi-target estimation. From Table 1 (right) and Fig. 3 (middle) we see that the agent outperforms the heuristic baselines as in the case with a single target. Automatic view selection termination does not yield as big improvements in accuracy as in the single-target case. In the single-target setting the agent stops early to avoid occluded and bad views, but when reconstructing all people there is more reason to keep selecting additional views to find some views which provide reasonable estimates for each person. This also explains the decreased gaps between the various methods – there may be many sets of cameras which together provide a fairly similar result when averaged over all people in the scene. A future improvement could include selective fusing a subset of estimates in each view. Running in auto mode still yields more accurate estimates than fixed

---

[5]We use SMPL (Loper et al. 2015) for the 3d shape models.

Figure 5: Visualizations of Pose-DRL reconstructing a given target person (red bounding box). Left: A *Mafia* test scene. The target is viewed from behind and is partially visible in the first view, producing the poor first estimate. As the agent moves to the next view, the person becomes more clearly visible, significantly improving the estimate. The last view from the front further increases accuracy. The agent decides to terminate after three views with error decreasing from 200.1 to 120.9 mm/joint. Right: An *Ultimatum* test scene where the agent only requires two viewpoints prior to automatically continuing to the next active-view. The target person is only partially visible in the initial viewpoint, and the right arm that is not visible results in a non-plausible configuration in the associated estimate. As the agent moves to the next viewpoint the person becomes fully visible, and the final fused estimate is both physically plausible and accurate. The reconstruction error reduces from 160 to 104 mm/joint.

schemes which use a larger number of views.

Runtimes. The runtimes[6] for Pose-DRL and baselines are shown in Fig. 3. DMHS and Faster R-CNN require 0.50 and 0.11 seconds per viewpoint, respectively, which constitutes the bulk of the processing time. The policy network has an overhead of about 0.01 seconds per action, negligible in relation to the pose estimation system.

Model	Settings	Maf	Ult	Pose
Pose-DRL	**full model**	144.7 (5)	157.5 (4)	135.1 (4)
	B^t **only**	153.5 (5)	166.9 (4)	134.4 (4)
	reset	152.5 (5)	160.8 (4)	133.4 (4)

Table 2: Ablations on the test sets, showing the effect of removing certain components of the DMHS-based Pose-DRL system. Results (errors, mm/joint) are for models that select a fixed number of views (shown in parentheses), where the number of views are the same as in Table 1. Providing more information than the base feature map B^t is crucial for crowded scenes with multiple people (*Maf*, *Ult*), as is including previous pose estimates in the current pose fusion.

5.2 Ablation Studies

In this section we compare the full agent to versions lacking parts of the model: i) providing only the base feature map B^t, and ii) not propagating the fused reconstruction $x_\star^t$ to the next active-view (*reset*), cf. (2). The results are given

---

[6]Shown for DMHS-based systems. Using MubyNet (which requires 1.01 seconds per image) gives runtime curves which look qualitatively similar.

in Table 2, and show that the full model outperforms the stripped-down versions for multi-people data (*Mafia*, *Ultimatum*), while simpler single-people data in *Pose* is not sensitive to removing some parts of the model. There is significantly more room for intelligent decision making for complex multi-people data, where the model has to avoid occlusions, and thus it requires a stronger state description and fusion approach. In contrast, selecting views in single-people scenes is less fragile to the particular camera choices as there is no risk of choosing views where the target is occluded.

6 Conclusions

In this paper we have presented *Pose-DRL*, a fully trainable deep reinforcement-learning based active vision model for human pose estimation. The agent has the freedom to move and explore the scene spatially and temporally, by selecting informative views that improve its accuracy. The model learns automatic stopping conditions for each moment in time, and transition functions to the next temporal processing step in video. We showed in extensive experiments – designed around the dense Panoptic multi-camera setup, and for complex scenes with multiple people – that Pose-DRL produces accurate estimates, and that our agent is robust with respect to the underlying pose estimator used. Moreover, the results show that our model learns to select an adaptively selected number of informative views which result in considerably more accurate pose estimates compared to strong multi-view baselines.

Practical developments of our methodology would include e.g. real-time intelligent processing of multi-camera video feeds or controlling a drone observer. In the latter case the model would further benefit from being extended to account for physical constraints, e.g. a single camera
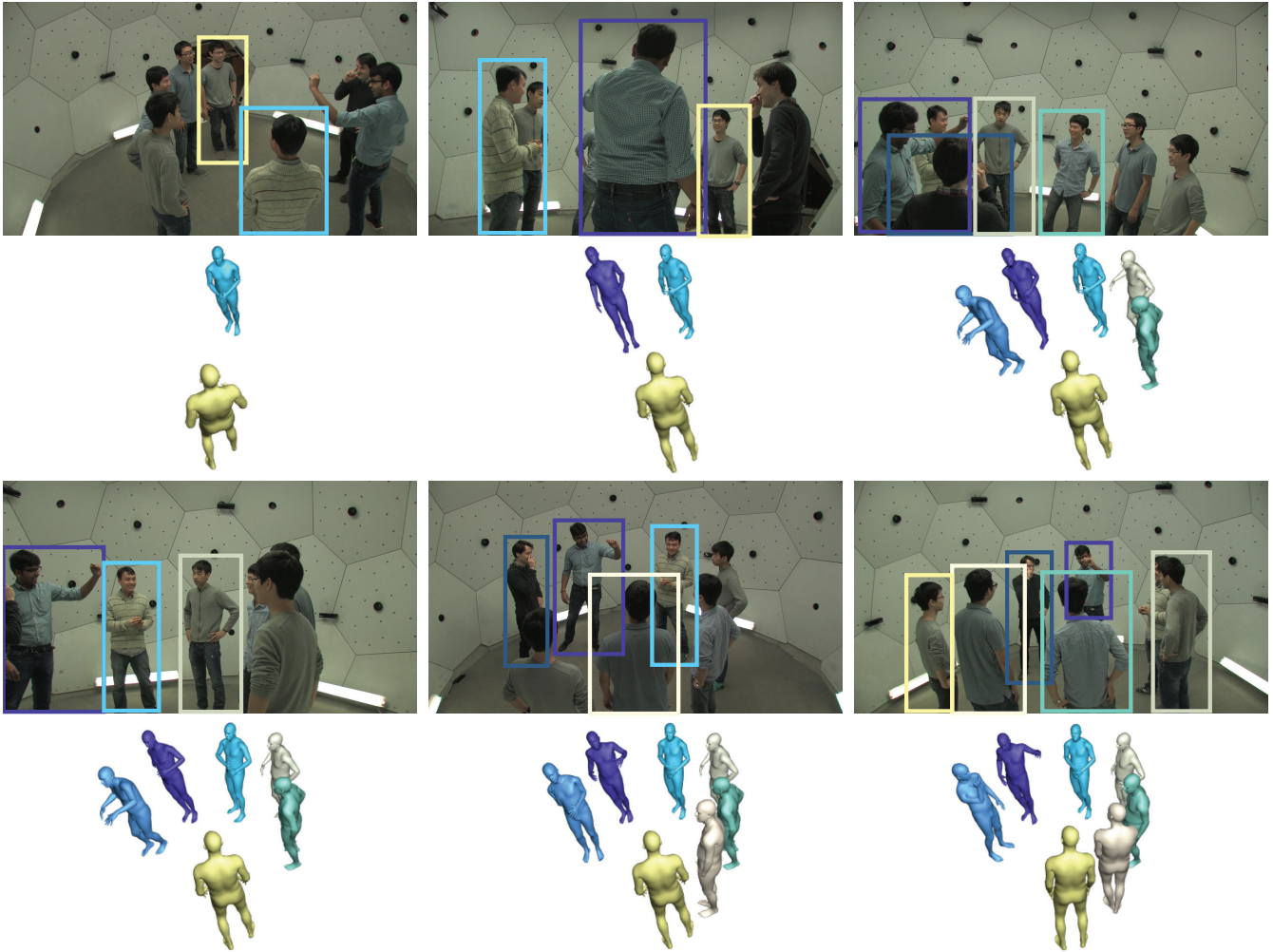
Figure 6: Visualization of how Pose-DRL performs multi-target pose estimation for an *Ultimatum* test scene. In this example the agent sees six viewpoints prior to automatically continuing to the next active-view. The mean error decreases from 358.9 to 114.6 mm/joint. Only two people are detected in the initial viewpoint, but the number of people detected increases as the agent inspects more views. Also, the estimates of already detected people improve as they get fused from multiple viewpoints.
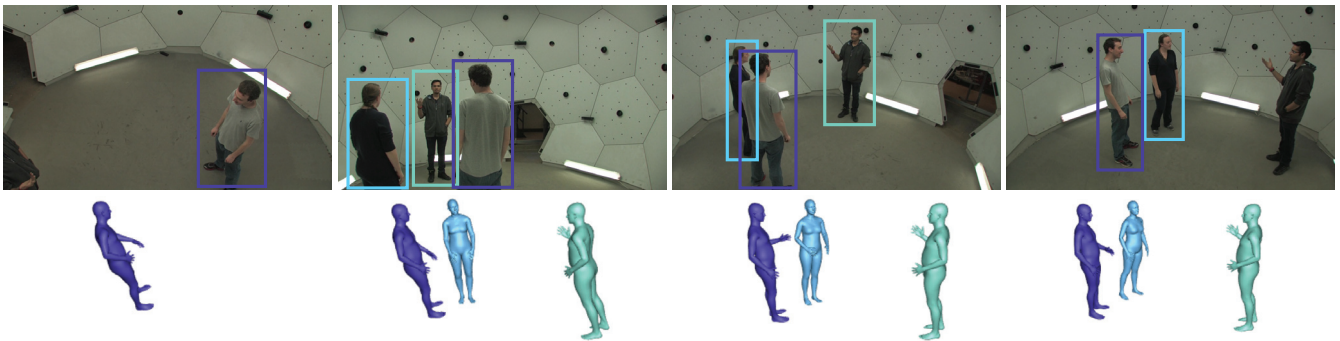

Figure 7: Visualization of how Pose-DRL performs multi-target pose estimation an *Ultimatum* validation scene. The agent chooses four viewpoints prior to automatically continuing to the next active-view. The mean error decreases from 334.8 to 100.9 mm/joint. Only one of the persons is visible in the initial viewpoint, and from a poor angle. This produces the first, incorrectly tilted pose estimate, but the estimate improves as the agent inspects more viewpoints. The two remaining people are successfully reconstructed in subsequent viewpoints.

and limited speed. Our paper is a key step since it presents fundamental methodology required for future applied research.

References

Ammirato, P.; Poirson, P.; Park, E.; Košecká, J.; and Berg, A. C. 2017. A dataset for developing and benchmarking active vision. In *ICRA*, 1378–1385. IEEE.

Bogo, F.; Kanazawa, A.; Lassner, C.; Gehler, P.; Romero, J.; and Black, M. J. 2016. Keep it SMPL: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*.

Caicedo, J., and Lazebnik, S. 2015. Active object localization with deep reinforcement learning. In *ICCV*.

Cheng, R.; Agarwal, A.; and Fragkiadaki, K. 2018. Reinforcement learning of active vision for manipulating objects under occlusions. In *CoRL*, 422–431.

Cheng, R.; Wang, Z.; and Fragkiadaki, K. 2018. Geometry-aware recurrent neural networks for active visual recognition. In *NeurIPS*, 5081–5091.

Das, A.; Kottur, S.; Moura, J. M.; Lee, S.; and Batra, D. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *CVPR*, 2951–2960.

Das, A.; Datta, S.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2018. Embodied question answering. In *CVPR*, volume 5, 6.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(7):1325–1339.

Jayaraman, D., and Grauman, K. 2018. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *CVPR*.

Johns, E.; Leutenegger, S.; and Davison, A. J. 2016. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, 3813–3822.

Joo, H.; Liu, H.; Tan, L.; Gui, L.; Nabbe, B.; Matthews, I.; Kanade, T.; Nobuhara, S.; and Sheikh, Y. 2015. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*.

Joo, H.; Simon, T.; and Sheikh, Y. 2018. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*.

Kanazawa, A.; Black, M. J.; Jacobs, D. W.; and Malik, J. 2018. End-to-end recovery of human shape and pose. In *CVPR*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. *ICLR*.

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A skinned multi-person linear model. *SIGGRAPH* 34(6):248:1–16.

Mehta, D.; Sridhar, S.; Sotnychenko, O.; Rhodin, H.; Shafiei, M.; Seidel, H.-P.; Xu, W.; Casas, D.; and Theobalt, C. 2017. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)* 36(4):44.

Papandreou, G.; Zhu, T.; Kanazawa, N.; Toshev, A.; Tompson, J.; Bregler, C.; and Murphy, K. 2017. Towards accurate multi-person pose estimation in the wild. In *CVPR*.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*.

Pavlakos, G.; Zhou, X.; and Daniilidis, K. 2018. Ordinal depth supervision for 3D human pose estimation. In *CVPR*.

Pirinen, A., and Sminchisescu, C. 2018. Deep reinforcement learning of region proposal networks for object detection. *CVPR*.

Pirinen, A.; Gärtner, E.; and Sminchisescu, C. 2019. Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction. In *NeurIPS*, 3907–3917.

Popa, A.-I.; Zanfir, M.; and Sminchisescu, C. 2017. Deep multitask architecture for integrated 2d and 3d human sensing. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Rhodin, H.; Robertini, N.; Casas, D.; Richardt, C.; Seidel, H.-P.; and Theobalt, C. 2016. General automatic human shape and motion capture using volumetric contour cues. In *ECCV*.

Rogez, G.; Weinzaepfel, P.; and Schmid, C. 2017. Lcr-net: Localization-classification-regression for human pose. In *CVPR*.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

von Marcard, T.; Henschel, R.; Black, M.; Rosenhahn, B.; and Pons-Moll, G. 2018. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional pose machines. In *CVPR*, 4724–4732.

Williams, R. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*.

Xia, F.; Zamir, A. R.; He, Z.; Sax, A.; Malik, J.; and Savarese, S. 2018. Gibson env: Real-world perception for embodied agents. In *CVPR*, 9068–9079.

Xiong, B., and Grauman, K. 2018. Snap angle prediction for 360 panoramas. In *ECCV*, 3–18.

Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; and Choi, J. Y. 2018. Action-driven visual object tracking with deep reinforcement learning. *IEEE transactions on neural networks and learning systems* 29(6):2239–2252.

Zanfir, A.; Marinoiu, E.; Zanfir, M.; Popa, A.-I.; and Sminchisescu, C. 2018. Deep network for the integrated 3d sensing of multiple people in natural images. In *NeurIPS*, 8410–8419.

Zanfir, A.; Marinoiu, E.; and Sminchisescu, C. 2018. Monocular 3d pose and shape estimation of multiple people in natural scenes–the importance of multiple scene constraints. In *CVPR*, 2148–2157.

Zhang, D.; Maei, H.; Wang, X.; and Wang, Y.-F. 2017. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*.

Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J. J.; Gupta, A.; Fei-Fei, L.; and Farhadi, A. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 3357–3364. IEEE.