UnrealEgo: A New Dataset for Robust Egocentric 3D Human Motion Capture

Hiroyasu Akada[1,2], Jian Wang[1], Soshi Shimada[1], Masaki Takahashi[2], Christian Theobalt[1], and Vladislav Golyanik[1]

[1] Max Planck Institute for Informatics, SIC
[2] Keio University

Abstract. We present *UnrealEgo*, *i.e.,* a new large-scale naturalistic dataset for egocentric 3D human pose estimation. UnrealEgo is based on an advanced concept of eyeglasses equipped with two fisheye cameras that can be used in unconstrained environments. We design their virtual prototype and attach them to 3D human models for stereo view capture. We next generate a large corpus of human motions. As a consequence, UnrealEgo is the first dataset to provide in-the-wild stereo images with the largest variety of motions among existing egocentric datasets. Furthermore, we propose a new benchmark method with a simple but effective idea of devising a 2D keypoint estimation module for stereo inputs to improve 3D human pose estimation. The extensive experiments show that our approach outperforms the previous state-of-the-art methods qualitatively and quantitatively. UnrealEgo and our source codes are available on our project web page[3].

Keywords: Egocentric 3D Human Pose Estimation, Naturalistic Data.

1 Introduction

Egocentric 3D human pose estimation has been actively researched recently [44,54,27,48,47,56,53,57]. Compared to cumbersome motion capture systems that require a fixed recording volume, the egocentric setup is more suitable to capture daily human activities in unconstrained environments. Example applications include XR technologies [19] and motion analysis for sport and health [40].

Several setup types were proposed for egocentric 3D human pose estimation. Some methods work on mobile devices such as a cap [54], a helmet [44] or a head-mounted display [48,47] equipped with a camera to capture egocentric views of a user's whole body. Although these methods show promising results, their setups are still not satisfactory for daily use; the cameras are mounted far from the user's body, which is inconvenient and restrictive. The recently introduced EgoGlass approach [57] tackles this issue by an eyeglasses-based setup with two cameras attached to the glasses frame. Their setup imposes fewer restrictions on users'

[3] `https://4dqv.mpi-inf.mpg.de/UnrealEgo/`

(a) Proposed concept of glasses equipped with two fisheye cameras (b) Human model wearing the glasses (c) Egocentric fisheye views

Fig. 1: Overview of the proposed UnrealEgo setup.

activities. We envision that with the recent development of smaller cameras [2] and smart glasses [4,6], the eyeglasses-based setup can be a de facto standard to capture daily human activities in various situations.

Along with that, there is a lack of datasets that would account for this new and advanced capture setting and that would allow developing algorithmic frameworks involving it. Furthermore, existing egocentric datasets are limited in several ways and cannot be easily re-purposed for 3D human pose estimation with the compact eyeglasses-based setup. First, the existing datasets do not contain complex human motions (such as breakdance and backflip) that are seen in daily human activities [44,54,48,57]. Second, the available egocentric datasets do not faithfully model the 3D environment [54,48]. Next, the existing stereo-based datasets [44,57] do not contain in-the-wild images. All in all, we note that there is no large-scale stereo-based dataset currently available. Consequently, a lack of a comprehensive and versatile egocentric dataset is a severely limiting factor in the development of methods for egocentric 3D perception.

To alleviate the issues mentioned above, we present *UnrealEgo*, *i.e.,* a new large-scale naturalistic and synthetic dataset for egocentric 3D human pose estimation. UnrealEgo is based on an advanced concept of an eyeglasses-based setup with two fisheye cameras symmetrically attached to the glasses frame. Fisheye cameras are getting more and more compact; they can capture a wider range of views than normal cameras which is beneficial for egocentric human pose estimation [44]. We use Unreal Engine [10] to synthetically design the eyeglasses as shown in Fig. 1-(a). We then attach the eyeglasses to realistic 3D human models (*RenderPeople*) [7] and capture in-the-wild stereo views in various 3D environments as shown in Fig. 1-(b), (c). Note that we prioritize the motion diversity in UnrealEgo. Fig. 2 shows examples of 3D human models in diverse poses from UnrealEgo. In total, UnrealEgo contains 450k in-the-wild stereo views (900k images in total) with the largest variety of motions among the existing egocentric datasets. UnrealEgo allows developing new methods that account for temporal

Fig. 2: Samples of characters and poses from UnrealEgo. We use 17 high-quality 3D RenderPeople models [7]. Also, we utilize Mixamo motions [5] and modify them to diversify our motion data. Please refer to our video for better visualizations and our supplementary asset list for characteristics of each human model.

changes of surrounding 3D environments (see Sec. 3) and evaluating the current state-of-the-art methods in highly challenging scenarios (see Sec. 5).

Furthermore, we propose a new benchmark approach that achieves state-of-the-art accuracy on UnrealEgo. At the core of our method is a heatmap-based 2D keypoint estimation module. It accepts stereo inputs and passes them to two weight-sharing encoders that produce feature maps in the latent space. The obtained feature maps are concatenated along with the channel dimensions and processed by a decoder that estimates 2D keypoint heatmaps (see Fig. 5). In extensive experiments, we observe that this simple but effective architecture brings significant improvements compared with existing methods [48,57] qualitatively and quantitatively by 13.5% on MPJPE and 14.65% on PA-MPJPE metrics.

In summary, the primary **contributions** of this work are as follows:

- *UnrealEgo*, *i.e.,* a new large-scale naturalistic dataset for egocentric 3D human motion capture.
- A new approach for 3D human pose estimation achieving state-of-the-art accuracy on the new benchmark dataset.

UnrealEgo is the first to provide 1) naturalistic in-the-wild stereo images with the largest variety of motions and 2) sequences with realistically and accurately-modeled changes of the surrounding 3D environments. This allows a more thorough evaluation of existing and upcoming methods for egocentric 3D vision, including the temporal component and global 3D poses.

2 Related Work

2.1 Datasets for Outside-in 3D Human Pose Estimation

Many datasets were proposed for 3D pose estimation with ground-truth annotations. Some of them are captured with optical markers [46,26,50], while the

others use marker-less mocap systems [37,36,28,55]. However, these datasets are mostly captured in the studio and usually lack the diversity of clothing, occlusions, and environments.

In the meantime, synthetic datasets have become popular because no costly mocap setups are required for annotations. Many such datasets are created by compositing people on background images [52,42,24,43,36,38]. Because of such composition, however, their images do not match real-world scenes in terms of the local pixel intensity statistics and distributions. Butler *et al.* [15] provide images rendered using underlying detailed 3D geometry and corresponding optical flows that can be used for tracking purposes. However, this dataset does not provide 3D joint annotations unlike ours.

The recent works by Zhu *et al.* [58] and Patel *et al.* [41] use 3D modeling tools and game engines [1,9,10] to render realistic images of rigged 3D human models in 3D environments. Unfortunately, these datasets are designed for outside-in pose estimation from an external camera viewpoint; they are not suitable and cannot be easily repurposed for egocentric 3D pose estimation.

2.2 Datasets for Egocentric 3D Human Pose Estimation

There exist several datasets specifically recorded for egocentric 3D human poses. Mo^2Cap^2 [54] is the first cap-based setup with a single wide-view fisheye camera attached 8cm away from the user. With this setup, Xu *et al.* [54] create a large-scale dataset by compositing SMPL models [33] on randomly-chosen backgrounds (real images), resulting in 530k images with 15 annotated keypoints per image. xR-EgoPose [48] approach uses a head-mounted display with a single fisheye camera equipped 2cm away from a user's nose. This work uses the Mixamo motion dataset [5] to animate 3D human models and renders egocentric views with HDR backgrounds with the help of the 3D rendering tool V-Ray [3]. Their dataset contains 380k photorealistic synthetic images with 25 body and 40 hand keypoints. However, both datasets contain only monocular images. They feature only simple (every-day) human motions (due to the restrictions imposed by their setups) and do not accurately model 3D environments and complex human trajectories in them. Hence, they do not cover most motions that can arise in egocentric 3D human pose estimation using a compact eyeglass-based setup. Ego4D [22] is a new large-scale dataset for egocentric vision. Unfortunately, it does not contain 3D annotations of human poses.

On the other hand, existing stereo egocentric datasets have several limitations. Rhodin *et al.* [44] proposed EgoCap, *i.e.,* a headgear with a pair of fisheye cameras equipped 25cm away from users to capture stereo views. Their dataset contains only 30k stereo image pairs with a limited variety of motions in a lab environment. More recently, EgoGlass [57] simplified the stereo setup with eyeglasses and two cameras equipped on the glasses frames. Although EgoGlass captured a relatively large-scale of images, *i.e.,* total 170k stereo pairs, *the dataset is captured only in a studio environment and is not publicly available.*

In contrast to existing datasets, UnrealEgo addresses the above shortcomings. Fig. 3 illustrates the differences among existing datasets and UnrealEgo. Firstly,

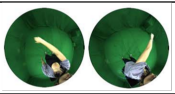
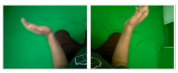
	Monocular Setting		Stereo Setting		
	Mo2Cap2[54]	xR-EgoPose[48]	EgoCap[44]	EgoGlass[57]	UnrealEgo
Device					
Example Data					
Distance to user's face	~8cm from the head	~2cm from the nose	~25cm from the head	~1cm from the head	~1cm from the head
Number of egocentric views	530k	380k	30k × 2 views	170k × 2 views	450k × 2 views
Number of keypoints	body: 15	body: 25 hand: 40	body: 17	body: 13	body: 32 hand: 40
Image generation	composite	composite	lab environments	lab environments	3D environments
Image quality	low	realistic	real	real	realistic
Motion diversity	middle	middle	low	low	high

Fig. 3: Comparison of datasets for egocentric 3D human pose estimation.

UnrealEgo provides stereo images in indoor and outdoor scenes. Secondly, it offers the largest number of images, *e.g.,* 15 times larger than EgoCap [44] and 2.5 times larger than EgoGlass [57]. Next, it contains naturalistic image sequences with accurately modeled geometry changes in the surrounding 3D environments. Also, it offers the largest number of body and hand keypoints. Furthermore, it is the most challenging egocentric dataset in terms of motion variety.

2.3 Methods for Egocentric 3D Human Pose Estimation

Existing methods for egocentric 3D human pose estimation can be divided into two groups in terms of egocentric settings. The first group aims at estimating 3D keypoints from monocular views. Mo2Cap2 [54] is the first CNN-based system to predict 3D poses. Tome *et al.* [48,47] follow a two-step approach using a multi-branch autoencoder to capture uncertainty in their predicted 2D heatmaps and to leverage rotation constraints [47]. Jiang *et al.* [27] predict 3D poses by utilizing the information of surrounding environments and extremities of the user's body. Zhang *et al.* [56] estimate 3D poses with fisheye distortions using an automatic calibration module. More recently, Wang *et al.* [53] proposed an optimization-based approach with a motion prior learned from an additional dataset for global 3D human motion capture. Even with their competitive results, these monocular methods often fail on complex motions (*e.g.,* due to the depth ambiguity).

The second group follows multi-view settings, including our work. EgoCap [44] is an optimization-based approach using a body-part detector and personalized 3D skeleton models. Cha *et al.* [17] developed a headset equipped with eight cameras; they introduced a CNN-based method to reconstruct a human body

Table 1: Comparison of human motion capture datasets.

Dataset	Subjects	Motions	Minutes	Dataset	Subjects	Motions	Minutes
ACCAD [11]	20	252	26.74	KIT [35]	55	4232	661.84
BMLhandball [31]	10	649	101.98	MPI HDM05 [39]	4	215	144.54
BMLmovi [21]	89	1864	174.39	MPI Limits [12]	3	35	20.82
BMLrub [49]	111	3061	522.69	MPI MoSh [32]	19	77	16.53
CMU [16]	96	1983	543.49	MPI-INF-3DHP [36]	8	-	-
D-FAUST [14]	10	129	5.73	SFU [51]	7	44	15.23
DanceDB [13]	20	151	203.38	SSM [34]	3	30	1.87
EKUT [35]	4	349	30.74	TCD Hands [25]	1	62	8.05
Eyes Japan [20]	12	750	363.64	TotalCapture [50]	5	37	41.1
Human3D [26]	11	-	-	Transitions [34]	1	110	15.1
Human4D [18]	8	148	72.60	AMASS [34]	344	11265	2420.86
HumanEva [46]	3	28	8.48	Ours	17	45520	3174.63

and an environment in 3D. EgoGlass [57] builds upon xR-EgoPose [48] and is one of the most accurate methods; its architecture contains two separate UNets for the stereo inputs in the 2D joint estimation module. In contrast to the reviewed works, this paper proposes a simple yet effective idea of devising a new 2D joint estimation module that accepts stereo inputs to significantly improve 3D pose estimation compared with the existing best-performing methods.

3 UnrealEgo Dataset

This section provides details of the UnrealEgo dataset, focusing on our setup, motions, and rendered egocentric data. Please also see our supplementary video for dynamic visualizations and our supplementary asset list.

3.1 Setup

We use Unreal Engine [10] to synthetically design the eyeglasses with two fisheye cameras equipped on the glasses frame as shown in Fig. 1-(a). The distance between the cameras is 12cm. The cameras' field of view amounts to 170°. We attach the glasses to 3D human models (RenderPeople) that perform different motions in various 3D environments. Fig. 1-(b) and (c) show an example of the human models in a Kyoto-inspired environment in Japan, and fisheye views.

Characters. We use 17 realistic RenderPeople 3D human models (commercially available) [7], nine female and eight male. These models are rigged and skinned based on the default 3D human skeleton of Unreal Engine [10]. Their skin color tones include pale white, white, light brown, moderate brown, dark brown, and black. Their clothing types include athletic pants, jeans, shorts, tights, dress pants, skirts, jackets, t-shirts, and long sleeves with diffident colors. Please see Fig. 2 for an overview of the 3D human models we use. Also, please see our supplement for detailed characteristics of each human model.

Table 2: Motion categories in our dataset.

Motion types	Motions	Minutes	Motion types	Motions	Minutes
1: jumping	1343	36.35	16: standing - whole body	3791	307.95
2: falling down	714	35.27	17: standing - upper body	5820	708.74
3: exercising	1225	82.07	18: standing - turning	1785	82.73
4: pulling	272	28.31	19: standing - to crouching	680	38.21
5: singing	1054	149.21	20: standing - forward	3417	93.68
6: rolling	136	4.69	21: standing - backward	1207	21.69
7: crawling	612	22.47	22: standing - sideways	1496	30.42
8: laying	612	30.92	23: dancing	5728	800.13
9: sitting on the ground	68	10.88	24: boxing	4012	160.53
10: crouching - normal	1802	127.90	25: wrestling	2958	119.63
11: crouching - turning	612	12.74	26: soccer	1892	69.63
12: crouching - to standing	850	29.46	27: baseball	476	27.31
13: crouching - forward	1020	29.50	28: basketball	272	7.54
14: crouching - backward	493	8.82	29: american football	85	6.07
15: crouching - sideways	646	11.69	30: golf	442	80.07

Motions. It is our top priority to include a wider variety of motions that can represent as many daily human activities as possible. Therefore, we first create a new large corpus of motions. Specifically, we utilize Mixamo motions [5] and modify them using Unreal Engine [10] to enhance their plausibility and diversify the motion data. We first manually fix some motions that involve self-penetration and then modify the motions in various ways, including the speed of motions, arm movements, foot stances, and head rotations. For further details, please refer to our supplement. In total, we created 45,520 natural motions for the 17 human models, *i.e.,* ≈2700 motions per model. We provide the details of our dataset in Tables 1 and 2. Table 1 compares existing mocap datasets and our motion data. Note that AMASS [34] is a collection of several existing motion capture datasets [11,31,49,21,16,14,13,35,20,18,46,35,39,12,32,51,25,50]. Our dataset contains the largest number of motions with the longest consecutive 3D human motions. Table 2 summarises the included motion categories.

3D Environments. We use 14 realistic 3D environments. They include a variety of indoor and outdoor scenes (*e.g.,* parks, roads, bridges, offices, gardens, playrooms, laboratories, cafeterias, trains, tennis courts, baseball fields, football fields, factories, European boulevards, North-American houses, Chinese rooms, Kyoto towns, and Japanese restaurants, at different times of day and night). Please see our supplementary asset list for further details.

Spawning Human Characters. It is important to create populated scenes to simulate real-world situations. To this end, we develop an algorithm to randomly place human models in 3D environments in Unreal Engine. As a preliminary step, we manually place K rectangles $B=\{B_1, ..., B_K\}$ where several human models can be spawned on even grounds. Here, let $S=\{S_1, ..., S_K\}$ be the areas of rectangles and $C=\{C_1, ..., C_K\}$ be their center positions in the world frame in Cartesian coordinate, respectively. As a first step, we choose i-th area $B_i \in B$ using area weighted probability $S_i / \sum_{i=1}^{K} S_i$. Secondly, we select T surrounding

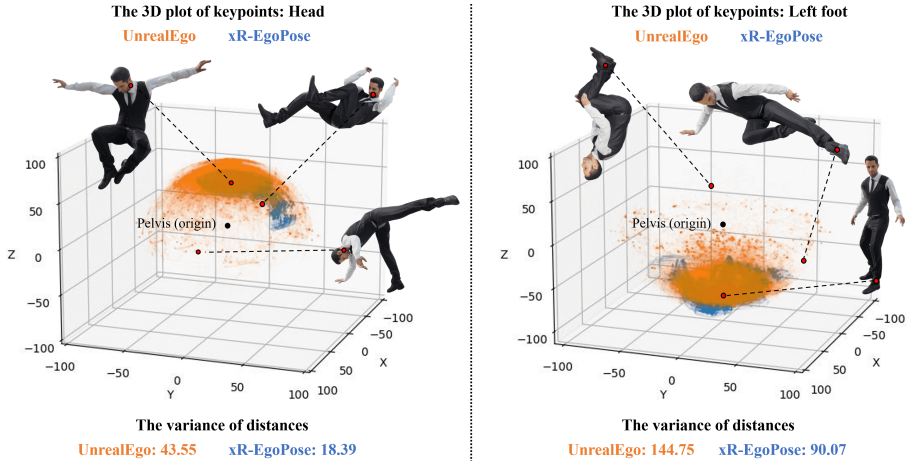


Fig. 4: Distributions of head and left foot locations in xR-EgoPose [48] (blue) and UnrealEgo (orange). The pelvis-relative 3D coordinates are on cm-scale.

rectangles $B_t \in B$, $t = \{1, \ldots, T\}$ with their center positions $C_t \in C$ being within 10m from C_i. Next, from all of the selected rectangles, we randomly sample world positions to place human models. The sampled positions are at least 1m far from each other. Lastly, we place human models by adjusting the heights of the lowest vertices of the human models to those of the sampled positions. About five models are spawned on average at once, and we render egocentric views from them. After that, we go back to the first step. This way, we randomly place the human models closer to each other in the 3D environment, and some rendered views can capture multiple models.

Rendering. We use a fisheye plugin [8] to render images until motions are completed, or a collision is detected. Here, we use the physics engine of Unreal Engine to detect collisions based on the pre-defined collision proxies (volumes) of the human models and the 3D environments. Around 100 stereo views per motion are rendered on average. The environments contain multiple light sources, including sky, points, and directional lights. Ray-tracing is enabled if the environments support it; rasterization rendering is used otherwise. Also, the rendering process of Unreal Engine includes deferred shading, global illumination, lit translucency, post-processing, and GPU particle simulation utilizing vector fields. Please refer to our supplement for more details on the asset rendering. All images are rendered on NVIDIA RTX 3090. The rendering speed is two frames per second.

3.2 Egocentric Dataset

We capture stereo fisheye images and depth maps with a resolution of 1024×1024 pixels each with 25 frames per second. Metadata is provided for each frame, including 3D joint positions, camera positions, and 2D coordinates of reprojected

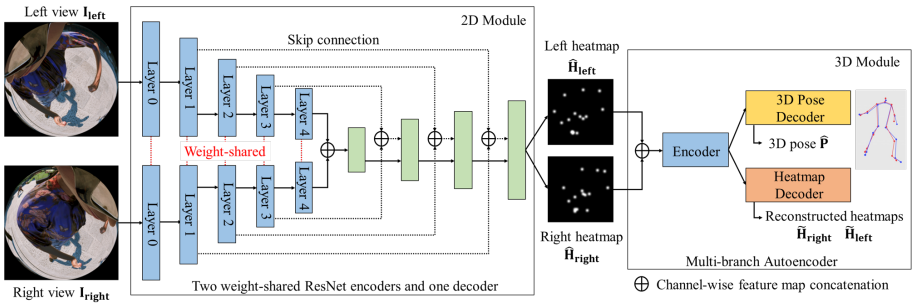


Fig. 5: Overview of the proposed method. Our network consists of a 2D module to predict 2D heatmaps of joint positions from stereo inputs (Sec. 4.1) and a 3D module to estimate 3D joint positions from the heatmaps (Sec. 4.2).

joint positions in the fisheye views. We randomly choose 10% of our motion data over all motion types, and capture 450k in-the-wild stereo views (900k images) in total. See Fig. 3 for the comparison with existing egocentric datasets.

As mentioned in Sec. 2.2, the motion variety is our top priority. UnrealEgo contains many complex motions in daily activities, some of which are difficult to capture with corresponding egocentric views in real-world settings. Example motions include breakdance and backflip in the dancing category shown in Table 2. To highlight the diversity of motions in UnrealEgo, we visualize the distributions of the keypoints in our UnrealEgo and xR-EgoPose [48] datasets in Fig. 4. Here, we use pelvis-relative 3D coordinates for head and left foot positions. Overall, the keypoints of UnrealEgo are more widespread with a larger variance of distances from the pelvis (origin) than those of xR-EgoPose. For example, in the left 3D plot of Fig. 4, the head is moving through a larger 3D space in UnrealEgo, even to areas below the pelvis, whereas head locations of xR-EgoPose are predominantly fixed above the pelvis. This shows that the UnrealEgo motions have a higher diversity of head positions.

4 Egocentric 3D Human Pose Estimation

In this section, we describe our egocentric 3D human pose estimation method. We firstly adopt a 2D module to predict 2D heatmaps of joint positions from stereo inputs and, next, a 3D module to generate 3D joint positions from the 2D heatmaps. Fig. 5 shows the overview of our network architecture. The main contribution of our method lies in the 2D module specifically designed for stereo inputs. This differs from the previous work [57], which uses two separate 2D modules for stereo views. In the following, we explain each module in detail.

Table 3: Comparisons on UnrealEgo with and w/o ImageNet pre-training.

Methods	Settings	MPJPE (σ)	PA-MPJPE (σ)
xR-EgoPose	Monocular	112.86 (1.16) / 123.15 (2.05)	88.71 (0.98) / 96.56 (1.27)
EgoGlass	Stereo	91.44 (0.84) / 107.70 (1.88)	70.21 (0.90) / 84.22 (0.99)
Ours	Stereo	**79.06 (0.25) / 87.31 (0.57)**	**59.95 (0.74) / 64.65 (0.93)**

4.1 2D Module

Our 2D module consists of two weight-sharing encoders and one decoder with unified skip connections [45] for stereo features as shown in Fig. 5. Here, we follow Zhao *et al.* [57] to use ResNet18 [23] as our encoder backbone. The 2D module takes stereo RGB images $\{\mathbf{I}_{\text{Left}}, \mathbf{I}_{\text{Right}}\} \in \mathbb{R}^{256\times256\times3}$ as inputs, and infers 2D joint locations represented as a set of heatmaps $\{\mathbf{H}_{\text{Left}}, \mathbf{H}_{\text{Right}}\} \in \mathbb{R}^{64\times64\times15}$. Here, we predict 15 joints in the neck, upper arms, lower arms, hands, thighs, calves, feet and balls of the feet. From each layer of the two weight-sharing encoders, we extract the features and concatenate them along the channel dimension. These features are then forwarded to corresponding decoder layers via skip connections. Unlike the 2D module of the previous work [57], our 2D module utilizes stereo information for heatmap estimation and, thus, boosts the performance of the 3D pose estimation task. For the training of the 2D module, we apply the mean squared error (mse) between the ground-truth heatmaps $\mathbf{H}_{\text{Left}}$ and $\mathbf{H}_{\text{Right}}$ and the estimated 2D heatmaps $\widehat{\mathbf{H}}_{\text{Left}}$ and $\widehat{\mathbf{H}}_{\text{Right}}$:

$$L_{\text{2D}} = \text{mse}(\mathbf{H}_{\text{Left}}, \widehat{\mathbf{H}}_{\text{Left}}) + \text{mse}(\mathbf{H}_{\text{Right}}, \widehat{\mathbf{H}}_{\text{Right}}). \tag{1}$$

4.2 3D Module

Following previous work [57], we adopt the same multi-branch autoencoder for our 3D module. Given the heatmaps $\widehat{\mathbf{H}}_{\text{Left}}$ and $\widehat{\mathbf{H}}_{\text{Right}}$ predicted by the 2D module as inputs, the 3D module firstly encodes them to get embedding features. These features are used in two decoder branches. The first branch is a 3D pose branch, which outputs the final 3D pose $\hat{\mathbf{P}} \in \mathbb{R}^{16\times3}$. Here, the number of output 3D joints is 16 as the head position is included. The second branch is a heatmap branch, which tries to reconstruct the predicted 2D heatmaps $\widetilde{\mathbf{H}}_{\text{Left}}$ and $\widetilde{\mathbf{H}}_{\text{Right}}$ so that the network can capture the uncertainty of the heatmaps.

Similar to [57], the overall loss function for the 3D module is as follows:

$$\begin{aligned} L_{\text{3D}} = \lambda_{\text{pose}}(\text{mpjpe}(\mathbf{P}, \hat{\mathbf{P}}) + \lambda_{\cos}\cos(\mathbf{P}, \hat{\mathbf{P}})) + \\ \lambda_{\text{hm}}(\text{mse}(\widehat{\mathbf{H}}_{\text{Left}}, \widetilde{\mathbf{H}}_{\text{Left}}) + \text{mse}(\widehat{\mathbf{H}}_{\text{Right}}, \widetilde{\mathbf{H}}_{\text{Right}})), \end{aligned} \tag{2}$$

where $\mathbf{P}$ is a ground-truth 3D pose, $\text{mpjpe}(\cdot)$ is the mean per joint position error and $\cos(\cdot)$ is a negative cosine similarity, *i.e.,*

Table 4: Quantitative evaluation on the general motions of UnrealEgo (MPJPE).

Methods	jumping	falling down	exercising	pulling,	singing	rolling	crawling	laying
xR-EgoPose	106.30	167.18	133.19	119.49	99.62	166.14	223.51	146.67
EgoGlass	88.55	135.25	105.11	89.96	75.54	143.64	199.27	114.85
Ours	**76.81**	**125.22**	**90.54**	**80.61**	**65.53**	**94.97**	**179.98**	**97.56**

Methods	sitting on the ground	crouching - normal	crouching - turning	crouching - to standing	crouching - forward	crouching - backward	crouching - sideways	standing - whole body
xR-EgoPose	274.99	172.25	173.77	108.96	119.95	136.52	145.81	94.34
EgoGlass	216.52	129.72	151.71	93.71	90.76	100.39	122.23	78.57
Ours	**195.28**	**120.65**	**131.82**	**81.28**	**76.04**	**81.31**	**88.54**	**67.67**

Methods	standing - upper body	standing - turning	standing - to crouching	standing - forward	standing - backward	standing - sideways	all
xR-EgoPose	93.36	103.28	101.60	99.72	105.86	114.28	112.61
EgoGlass	76.83	84.12	82.03	82.96	85.15	93.61	91.27
Ours	**65.92**	**74.55**	**73.21**	**70.86**	**70.40**	**79.06**	**79.57**

Table 5: Quantitative evaluation on the sports motions of UnrealEgo (MPJPE).

Methods	dancing	boxing	wrestling	soccer	baseball	basketball	american football	golf	all
xR-EgoPose	116.75	97.33	116.65	104.65	103.75	98.65	149.76	117.50	113.28
EgoGlass	95.37	77.66	96.63	88.30	93.60	74.31	118.34	79.35	91.71
Ours	**79.86**	**69.34**	**84.02**	**76.54**	**74.27**	**62.09**	**103.79**	**72.06**	**78.19**

$$\mathrm{mpjpe}(\mathbf{P}, \hat{\mathbf{P}}) = \frac{1}{BJ} \sum_{i=1}^{B} \sum_{j=1}^{J} ||\mathbf{P}_i^j - \hat{\mathbf{P}}_i^j||_2, \tag{3}$$

$$\cos(\mathbf{P}, \hat{\mathbf{P}}) = -\frac{1}{B} \sum_{i=1}^{B} \sum_{l=1}^{L} \frac{\mathbf{P}_i^l \cdot \hat{\mathbf{P}}_i^l}{||\mathbf{P}_i^l||\,||\hat{\mathbf{P}}_i^l||}, \tag{4}$$

where B is the batch size, J is the number of joints, L is the number of limbs, and $\mathbf{P}_i^l \in \mathbb{R}^3$ is the l-th bone of the human skeleton.

5 Experiments

5.1 Implementation Details

We randomly split UnrealEgo into 3,821 motions (357,317 stereo views) for training, 494 motions (46,207 stereo views) for validation, and 526 motions (48,080 stereo views) for testing. The input images and ground-truth 2D heatmaps are resized to 256×256 and 64×64, respectively. The 2D module and the 3D module are trained separately on a Quadro RTX 8000 with a batch size of 16. We set the hyper-parameters as $\lambda_{\mathrm{pose}} = 0.1$, $\lambda_{\cos} = 0.01$, and $\lambda_{\mathrm{hm}} = 0.001$. The modules

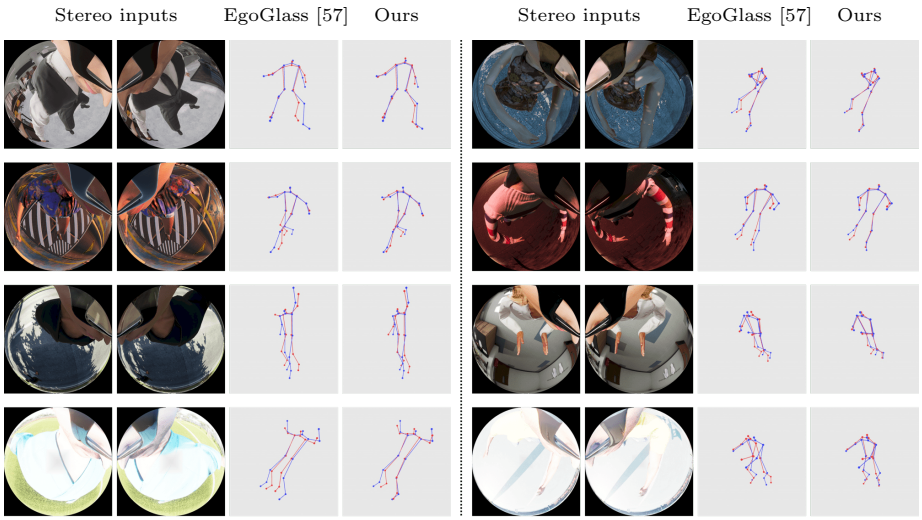


Fig. 6: Qualitative results on UnrealEgo (blue: ground truth; red: prediction).

are trained with Adam optimizer [30] for ten epochs, starting with a learning rate of 10^{-3} for the first half epochs and applying a linearly decaying rate for the next half. We perform the experiments three times and report average scores and standard deviations (denoted by σ).

5.2 Comparisons

As our comparison methods, we adopt state-of-the-art methods for egocentric 3D human pose estimation, *i.e.,* EgoGlass [57] and xR-EgoPose [48]. Since their source codes are not available, we re-implement and tailor them for UnrealEgo. We train xR-EgoPose on the left views of UnrealEgo. For the sake of evaluation under the same conditions, we remove a body part branch with segmentation supervision in EgoGlass as xR-EgoPose does not use it. We follow the previous works and report the Mean Per Joint Position Error (MPJPE) and the Mean Per Joint Position Error with Procrustes alignment [29] (PA-MPJPE). Here, Procrustes alignment finds optimal rigid transformation and scale between the predicted and ground-truth 3D poses.

5.3 Results

We present results on the UnrealEgo test sequence. Table 3 quantitatively evaluates our approach and competing methods with and without ImageNet pre-training for the encoder. Overall, our method outperforms the previous best-performing method [57], across all metrics for both experiments with and without ImageNet. Specifically, our method with the pre-trained encoder shows significant improvement by 13.5% on MPJPE and 14.65% on PA-MPJPE compared

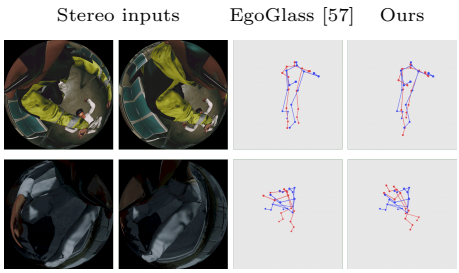


Fig. 7: Qualitative results for failure cases on UnrealEgo.

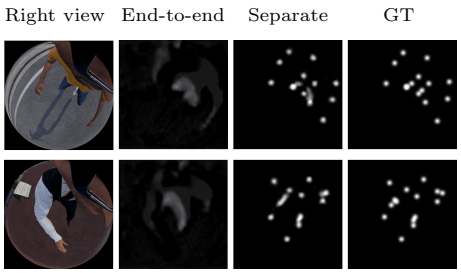


Fig. 8: Heatmap estimation results with two different training strategies.

Table 6: Ablation study for the backbone of the 2D heatmap module.

Backbones	MPJPE (σ)	PA-MPJPE (σ)
ResNet18	**79.06 (0.25)**	**59.95 (0.74)**
ResNet34	80.50 (0.78)	60.04 (0.60)
ResNet50	80.07 (0.45)	60.08 (0.63)
ResNet101	80.15 (0.06)	60.57 (0.79)

Table 7: Ablation study for the weight sharing in the 2D heatmap module.

Backbones	MPJPE (σ)	PA-MPJPE (σ)
weight sharing	**79.06 (0.25)**	**59.95 (0.74)**
no weight sharing	83.54 (1.30)	62.29 (0.45)

to EgoGlass [57]. All methods, including ours, benefit from the ImageNet pre-training; the performance of our approach is boosted by 9.4% on MPJPE and 7.2% on PA-MPJPE.

We also break down the test sequence into 30 motion types as shown in Table 4 for general motions and Table 5 for sports motions. Both tables indicate that our method achieves significant superiority for all motion types. See Fig. 6 for the qualitative results. Even with the occlusions and complex poses in various environments, our method estimates the 3D poses much better than EgoGlass.

It is also worth analyzing failure cases. According to Table 4, bending motions (such as sitting on the ground or crouching) are reconstructed with comparably low accuracy. This is because the lower body parts are occluded by the upper body, especially when people crouch down as shown in Fig. 7. Even with the stereo inputs, these methods still can not perform well on some motions that are occasionally seen in daily human activities.

5.4 Ablation Study

We first ablate different encoder backbone architectures for our 2D module in Table 6. All variants generate the heatmap with the same resolution and the 3D module shares the same architecture. The experiment suggests that all of the models yield similar results but at a higher computational cost for a larger backbone. For example, the difference between ResNet18 and Resnet50 is only 0.2% on PA-MPJPE. This result is also observed in the previous work [47], showing that a larger backbone does not necessarily lead to performance improvements.

Table 8: Ablation study on the training strategy.

Backbones	MPJPE (σ)	PA-MPJPE (σ)
Separate training	**79.06 (0.25)**	**59.95 (0.74)**
End-to-end training	80.67 (0.58)	61.72 (0.55)

Next, we show the effect of weight sharing in the encoder backbone of our 2D keypoint estimation module in Table 7. The weight-sharing backbone performs better than the encoder without weight sharing by 5.4% on MPJPE and 3.8% on PA-MPJPE. One possible reason for this result is that the weight-sharing backbone can see more views during training, leading to a better feature extractor. Therefore, we use the weight-sharing strategy for all experiments.

Lastly, we conduct the experiment with different training strategies, *i.e.,* separate training and end-to-end training for our 2D keypoint estimation and 3D estimation module, as shown in Table 8. The result indicates that the separate training yields slightly better performance than the end-to-end training by 2.0% on MPJPE and 2.9% on PA-MPJPE. We also visualize the heatmaps predicted by our network with the different training strategies in Fig. 8. It is interesting to note that separate training leads to relatively accurate heatmap estimation while the network trained in an end-to-end manner tries to capture the whole body. Although this visual result can change depending on the hyper-parameters, we follow the same hyper-parameter setting in the previous work [57] and choose the separate training strategy for all experiments.

6 Conclusions

We presented *UnrealEgo*, *i.e.,* a new large-scale naturalistic dataset for egocentric 3D human pose estimation. It allows a comprehensive evaluation of existing and upcoming methods for egocentric 3D vision, including the temporal component and global 3D poses. Our simple yet effective architecture for egocentric 3D human pose estimation brings significant improvement compared to previous best-performing methods qualitatively and quantitatively. In addition, our extensive ablation studies validate our architectural design choices for the stereo inputs and the training strategy. Although our method achieved state-of-the-art results, there are still failure cases due to occlusions and complex motions. In future work, we are interested in incorporating explicit 3D geometry obtained from our stereo fisheye setup for further performance improvements.

Acknowledgements. We thank Silicon Studio Corp. for providing the fisheye plugin. Hiroyasu Akada and Masaki Takahashi were supported by the Core Research for Evolutional Science and Technology of the Japan Science and Technology Agency (JPMJCR19A1). Jian Wang, Soshi Shimada, Vladislav Golyanik and Christian Theobalt were supported by the ERC Consolidator Grant 4DReply (770784).

References

1. Blender (2022), https://www.blender.org
2. Calicam fisheye stereo camera (2022), https://astar.ai/products/stereo-camera
3. Chaos v-ray (2022), https://www.chaos.com/
4. glass (2022), https://www.google.com/glass/start/
5. Mixamo (2022), https://www.mixamo.com
6. Ray-ban stories smart glasses (2022), https://www.ray-ban.com/usa/ray-ban-stories
7. Renderpeople (2022), https://renderpeople.com
8. Siliconstudio (2022), https://www.siliconstudio.co.jp/en/
9. Unity (2022), https://unity.com
10. Unreal engine (2022), https://www.unrealengine.com
11. Advanced Computing Center for the Arts and Design: ACCAD MoCap Dataset, `https://accad.osu.edu/research/motion-lab/mocap-system-and-data`
12. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3D human pose reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
13. Aristidou, A., Shamir, A., Chrysanthou, Y.: Digital dance ethnography: Organizing large dance collections. J. Comput. Cult. Herit. **12**(4) (2019)
14. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: Computer Vision and Pattern Recognition (CVPR) (2017)
15. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conf. on Computer Vision (ECCV) (2012)
16. Carnegie Mellon University: CMU MoCap Dataset, `http://mocap.cs.cmu.edu`
17. Cha, Y.W., Price, T., Wei, Z., Lu, X., Rewkowski, N., Chabra, R., Qin, Z., Kim, H., Su, Z., Liu, Y., Ilie, A., State, A., Xu, Z., Frahm, J.M., Fuchs, H.: Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. IEEE Transactions on Visualization and Computer Graphics **24**(11), 2993–3004 (2018)
18. Chatzitofis, A., Saroglou, L., Boutis, P., Drakoulis, P., Zioulis, N., Subramanyam, S., Kevelham, B., Charbonnier, C., Cesar, P., Zarpalas, D., et al.: Human4d: A human-centric multimodal dataset for motions and immersive media. IEEE Access **8**, 176241–176262 (2020)
19. Elgharib, M., Mendiratta, M., Thies, J., Nie/ssner, M., Seidel, H.P., Tewari, A., Golyanik, V., Theobalt, C.: Egocentric videoconferencing. ACM Transactions on Graphics **39**(6) (2020)
20. Eyes JAPAN Co. Ltd.: Eyes Japan MoCap Dataset, `http://mocapdata.com`
21. Ghorbani, S., Mahdaviani, K., Thaler, A., Kording, K., Cook, D.J., Blohm, G., Troje, N.F.: Movi: A large multi-purpose human motion and video dataset. PLOS ONE **16**(6), 1–15 (2021)
22. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Computer Vision and Pattern Recognition (CVPR) (2022)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (2016)
24. Hoffmann, D.T., Tzionas, D., Black, M.J., Tang, S.: Learning to train with synthetic humans. In: German Conference on Pattern Recognition (GCPR) (2019)

25. Hoyet, L., Ryall, K., McDonnell, R., O'Sullivan, C.: Sleight of hand: Perception of finger motion from reduced marker sets. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. p. 79–86. I3D '12 (2012)
26. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: 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 (2014)
27. Jiang, H., Ithapu, V.K.: Egocentric pose estimation from human vision span. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
28. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: International Conference on Computer Vision (ICCV) (2015)
29. Kendall, D.G.: A Survey of the Statistical Theory of Shape. Statistical Science **4**(2), 87 – 99 (1989)
30. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) (2015)
31. Lab, B.M.: BMLhandball Motion Capture Database, `https://www.biomotionlab.ca//`
32. Loper, M., Mahmood, N., Black, M.J.: MoSh: Motion and Shape Capture from Sparse Markers. ACM Trans. Graph. **33**(6) (2014)
33. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (2015)
34. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
35. Mandery, C., Terlemez, O., Do, M., Vahrenkamp, N., Asfour, T.: The KIT whole-body human motion database. In: International Conference on Advanced Robotics (ICAR) (2015)
36. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: International Conference on 3D Vision (3DV) (2017)
37. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. In: International Conference on 3D Vision (3DV) (2018)
38. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. In: 3D Vision (3DV), 2018 Sixth International Conference on (2018)
39. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation mocap database HDM05. Tech. Rep. CG-2007-2 (2007)
40. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Egocentric vision-based action recognition: A survey. Neurocomputing **472**, 175–197 (2022)
41. Patel, P., Huang, C.H.P., Tesch, J., Hoffmann, D.T., Tripathi, S., Black, M.J.: AGORA: Avatars in geography optimized for regression analysis. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
42. Pumarola, A., Sanchez, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: Modeling the Geometry of Dressed Humans. In: International Conference in Computer Vision (ICCV) (2019)
43. Ranjan, A., Hoffmann, D.T., Tzionas, D., Tang, S., Romero, J., Black, M.J.: Learning multi-human optical flow. International Journal of Computer Vision (IJCV) (128), 873–890 (2020)

44. Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.P., Schiele, B., Theobalt, C.: Egocap: egocentric marker-less motion capture with two fisheye cameras. ACM Transactions on Graphics (TOG) **35**(6), 1–11 (2016)
45. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention (2015)
46. Sigal, L., Balan, A., Black, M.J.: HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International Journal of Computer Vision (IJCV) **87**(4), 4–27 (2010)
47. Tomè, D., Alldieck, T., Peluse, P., Pons-Moll, G., de Agapito, L., Badino, H., la Torre, F.D.: Selfpose: 3d egocentric pose estimation from a headset mounted camera. IEEE transactions on pattern analysis and machine intelligence **PP** (2020)
48. Tome, D., Peluse, P., Agapito, L., Badino, H.: xr-egopose: Egocentric 3d human pose from an hmd camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
49. Troje, N.F.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. Journal of Vision **2**(5) (2002)
50. Trumble, M., Gilbert, A., Malleson, C., Hilton, A., Collomosse, J.: Total capture: 3d human pose estimation fusing video and inertial sensors. In: British Machine Vision Conference (BMVC) (2017)
51. University, S.F., of Singapore, N.U.: SFU Motion Capture Database, `http://mocap.cs.sfu.ca/`
52. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
53. Wang, J., Liu, L., Xu, W., Sarkar, K., Theobalt, C.: Estimating egocentric 3d human pose in global space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
54. Xu, W., Chatterjee, A., Zollhoefer, M., Rhodin, H., Fua, P., Seidel, H.P., Theobalt, C.: Mo^2Cap^2 : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. IEEE Transactions on Visualization and Computer Graphics (2019)
55. Yu, Z., Yoon, J.S., Lee, I.K., Venkatesh, P., Park, J., Yu, J., Park, H.S.: Humbi: A large multiview dataset of human body expressions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
56. Zhang, Y., You, S., Gevers, T.: Automatic calibration of the fisheye camera for egocentric 3d human pose estimation from a single image. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2021)
57. Zhao, D., Wei, Z., Mahmud, J., Frahm, J.M.: Egoglass: Egocentric-view human pose estimation from an eyeglass frame. In: International Conference on 3D Vision (3DV) (2021)
58. Zhu, T., Karlsson, P., Bregler, C.: Simpose: Effectively learning densepose and surface normals of people from simulated data. In: European Conference on Computer Vision (ECCV) (2020)