Robust Robot Motion Retargeting: Rig Unification and Application to Diverse Robots

Taemoon Jeong
Korea University
Taehyun Byun
Korea University
Jihoon Kim
CINAMON
Keunjoon Choi
Rainbow Robotics
Jaesung Oh
Rainbow Robotics
Sungpyo Lee
Naver Labs
Omar Darwish
University of Illinois Urbana-Champaign
Joohyung Kim
University of Illinois Urbana-Champaign
Sungjoon Choi
sungjoon-choi@korea.ac.kr

Korea University

Article

Keywords: Robot Motion Retargeting, Humanoid Robots, Trajectory Optimization, Expressive Robot Motion Generation

Posted Date: July 10th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4544618/v1

Additional Declarations: No competing interests reported.

Robust Robot Motion Retargeting: Rig Unification and Application to Diverse Robots

Taemoon Jeong[1], Taehyun Byun[1], Jihoon Kim[2], Keunjoon Choi[3], Jaesung Oh[3], Sungpyo Lee[4], Omar Darwish[5], Joohyung Kim[5], Sungjoon Choi[1*]

[1*]Department of Artificial Intelligence, Korea University, Seoul, Republic of Korea.
[2]CINAMON, Seoul, Republic of Korea.
[3]Rainbow Robotics, Daejeon, Republic of Korea.
[4]NAVER LABS, Seongnam, Gyeonggi-do, Republic of Korea.
[5]Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Champaign, IL, USA.

*Corresponding author(s). E-mail(s): sungjoon-choi@korea.ac.kr;
Contributing authors: taemoon-jeong@korea.ac.kr;
taehyun-byun@korea.ac.kr; jihoon@cinamon.io;
keunjun.choi@rainbow-robotics.com; jsoh91@rainbow-robotics.com;
s.p.lee@naverlabs.com; okd2@illinois.edu; joohyung@illinois.edu;

Abstract

Humanoid robots are increasingly being developed for seamless interaction with humans in various domains. However, generating expressive and feasible motions for these robots remains a significant challenge due to their complex kinematic structures and physical constraints. We propose a robust and automated pipeline for motion retargeting that enables the generation of natural motions for diverse humanoid robots using various motion data sources. Our approach unifies different kinematic configurations into a single predefined rig and refines the motion trajectory, considering factors such as balance and contact. The retargeted motion is then fine-tuned to closely follow the source motion while adhering to the robot's physical limits. We demonstrate the effectiveness of our methodology through successful applications on 12 simulated robots and validation on three real robots. This work represents a significant step towards automating expressive motion generation for humanoid robots, enabling their deployment in various real-world scenarios.

Keywords: Robot Motion Retargeting, Humanoid Robots, Trajectory Optimization, Expressive Robot Motion Generation

1 Introduction

Humanoid robots have undergone significant advancements in recent years, with various research institutions and companies developing increasingly sophisticated platforms [1, 2]. These robots are designed to interact with humans and perform complex tasks in a variety of settings, from industrial environments to entertainment and education. As humanoid robots become more prevalent and integrated into our daily lives, the importance of generating natural and expressive motions for these robots has become increasingly evident [3].

However, creating natural and expressive motions for humanoid robots remains a challenging task. Generating robot motions requires careful consideration of the robot's physical characteristics and limitations, as well as the desired expressiveness and aesthetics of the movements [4]. When performed manually, this process can be time-consuming and labor-intensive. This highlights the need for the development of efficient and automated methods for generating humanoid robot motions.

Motion retargeting effectively generates expressive motions for robots with different morphologies [5–7]. This method enables the efficient transfer of human or animated character motions to target robots, making it crucial for various applications. In human-robot interaction and assistive robotics, motion retargeting allows robots to perform natural and expressive movements, enhancing their ability to communicate and interact with humans [8, 9]. Furthermore, this approach opens up new opportunities in the entertainment industry by enabling robots to participate in programs such as plays and musicals, where they can showcase human-like movements and emotions.

Robot motion retargeting is the process of transferring motion data from one source, such as a human or an animated character, to different robots with distinct morphologies or kinematic structures [10]. This process involves transferring motion data from various sources, such as motion capture data and RGB videos, to generate realistic and adaptable robot motions. By enabling human-like movements and enhancing flexibility, robot motion retargeting plays a crucial role in adapting motions to robots with different kinematic structures and capabilities, making it an essential tool for creating natural and expressive robot behaviors.

There are two main approaches to robot motion retargeting: Joint (configuration) space and Task (Cartesian) space formulations. Joint space approaches involve manually mapping human joint values to target robots, taking into account the robot's kinematic structure. For example, Safonova et al. [11] constrained upper body gestures to a Sarcos humanoid robot, while Yamane et al. [12, 13] and Ayusawa and Yoshida [14] focused on whole-body control and retargeting motions with geometric considerations, respectively. On the other hand, Task space formulation emphasizes the positions and orientations of end-effectors in Cartesian space, allowing for more human-like motion resemblance without direct joint value mapping. Dariush et al. [15] introduced markerless retargeting with the Honda humanoid robot, ASIMO, and Bin

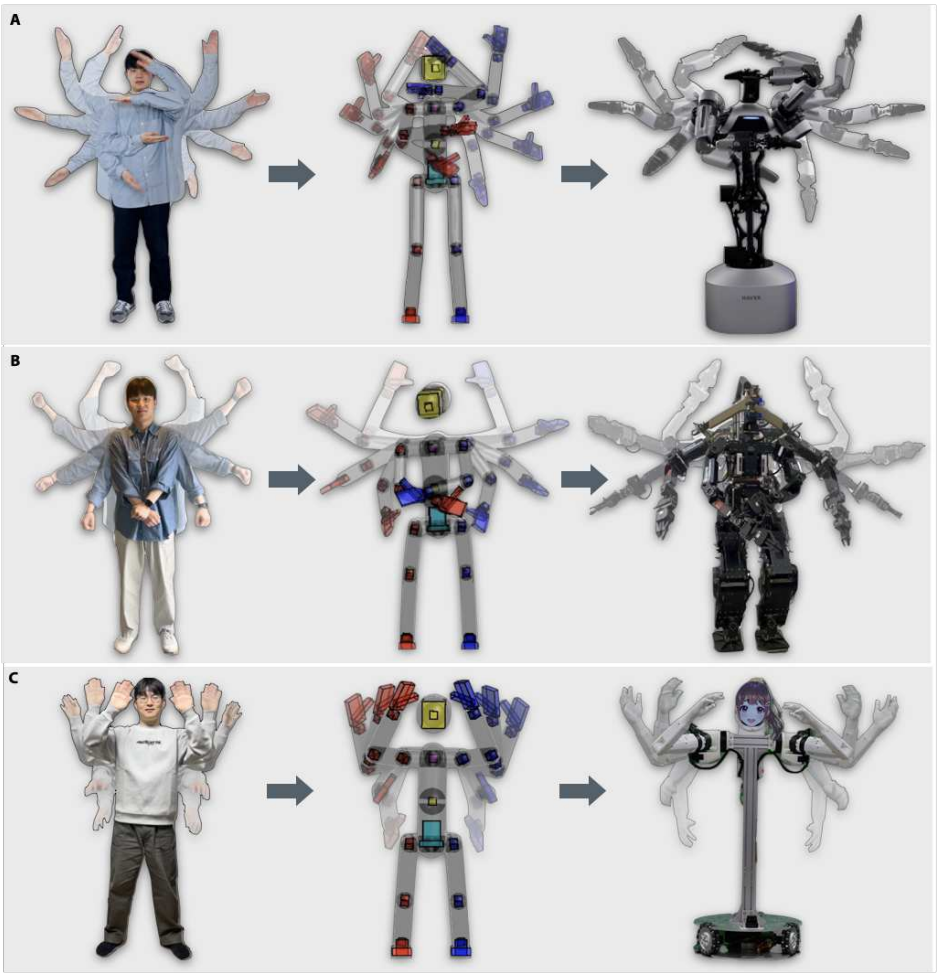

Fig. 1 Motion retargeting of videos taken from mobile phones to real robots. (**A**) The retargeted AMBIDEX motion. (**B**) The retargeted THORMANG motion. (**C**) The retargeted JF2 motion.

Hammam et al. [16] proposed a kinodynamically consistent method using task points. Moreover, Darvish et al. [17] presented a teleoperation framework for humanoid robots, applying a novel retargeting approach based on inverse kinematics in the iCub robot's task space.

Real-time motion transfer has also been a focus of research. Dariush et al. [18] and L.Penco et al. [19] proposed frameworks for online task space retargeting and real-time whole-body motion transfer, respectively, addressing joint limit, velocity, and self-collision constraints. Khalil et al. [20] developed a motion retargeting framework for the humanoid robot Pepper using a single-view camera and human pose estimation. They emphasized the need for noise reduction in pose estimation as a direction for future work to improve the accuracy and robustness of the retargeting process.

Data-driven approaches have gained significant attention in motion retargeting research [21–24]. For example, Yamane et al. [25] used Gaussian Process Latent Variable Models (GPLVM) [26] to generate motions for non-humanoid characters from human motion data. GPLVM creates a shared latent space between the source and target motions. However, ensuring the feasibility of the generated robot motions remains a challenge. Kim et al. [27] developed a cyclic optimization method based on deep reinforcement learning, utilizing human-in-the-loop feedback to enhance the motion retargeting policy. Choi et al. [28, 29] proposed semi-supervised and self-supervised learning methods, such as LWL2 and S3LE, to generate natural and feasible robot motions while avoiding collisions and adhering to position limits. These methods leverage paired data of human poses and robot configurations to learn a shared latent space for motion retargeting.

In this paper, we introduce a robust and enhanced method for automated robot motion retargeting. This method can effectively handle noise and physically-implausible poses in diverse source motion data, including motion capture and pose estimation data obtained from RGB videos, facilitating the transfer to a wide range of target robot hardware (Fig. 1). We consider two types of motion data: motion capture data and pose estimation data. Motion capture data typically acquired using expensive devices such as VICON [30] or OptiTrack [31], offer high-quality motion information but vary in skeleton structure depending on the device.

Pose estimation methods [32–36] can estimate 3D human poses from RGB videos without requiring depth information. However, the lack of depth information causes estimating 3D poses from RGB images to be an ill-posed problem. Furthermore, factors like occlusion, camera viewpoint, and background clutter can affect these estimations, leading to inaccuracies in the motion data, such as inclinations or jitters. Dariush et al. [18] and Khalil et al. [20] have highlighted the issue of inaccuracy in pose estimation methods. Therefore, refining inaccurate and noisy motion data into feasible motion is critical for the success of motion retargeting methods.

We propose a common-rigging process to handle noisy motion data with different skeletons (Fig.2B). The process consists of a pre-rigging step and a post-rigging step. The pre-rigging step aims to unify different human skeletons (Fig. 3A) into a single predefined rig, called common-rig (Fig. 3B). Unlike the source skeleton, which is typically meshless, the common-rig includes a rigid body structure and physical properties such as volume, mass, and moment of inertia for each link, enabling effective handling of self-collisions and noisy poses. The post-rigging step refines noisy motions to feasible motions by considering the physical properties of the common-rig and ensuring feet are planted on the ground. Through the common-rigging process, we enable the utilization of diverse and potentially noisy motion sources, including various motion capture formats or estimated poses extracted from RGB videos.

Next, we propose a robot motion retargeting process that transfers rig-unified motions to various robots with different kinematic structures and physical constraints, such as joint angle limits and velocity-acceleration bounds (Fig. 2C). Our approach transfers the motion of the common-rig to the robot by computing the target pose in the robot's task space using a direction vector-based method and solving an inverse kinematics problem to obtain the corresponding joint angles. Then, it optimizes the

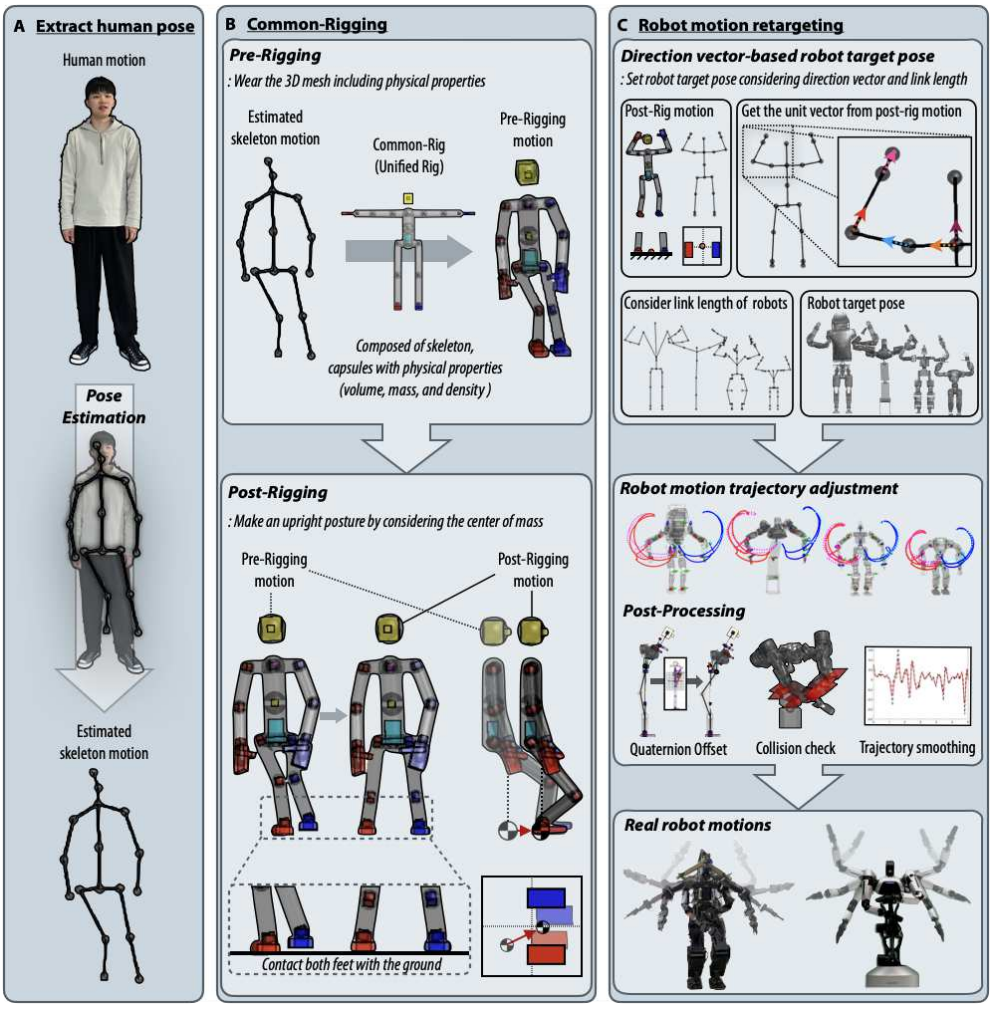

Fig. 2 Overview of proposed method. (**A**) Extracting human skeleton motion from motion capture systems or pose estimations. (**B**) Common-rigging process, which includes optimization-based methods called pre-rigging and post-rigging, aims to unify different human skeleton structures into a single predefined rig, referred to as a common-rig. (**C**) Robot motion retargeting consists of Direction vector-based robot target pose, Robot motion trajectory adjustment, and Post-processing. The process generates feasible real robot motions that preserve the features of the source human motion.

robot's motion trajectories to closely follow the reference trajectories while adhering to robot-specific constraints and avoiding self-collisions. Finally, the optimized robot motion is smoothed to ensure dynamic feasibility. In summary, the main contributions of this work are:

1. An optimization-based common-rigging process that unifies diverse motion data into a standardized representation and refines noisy, physically-implausible motions

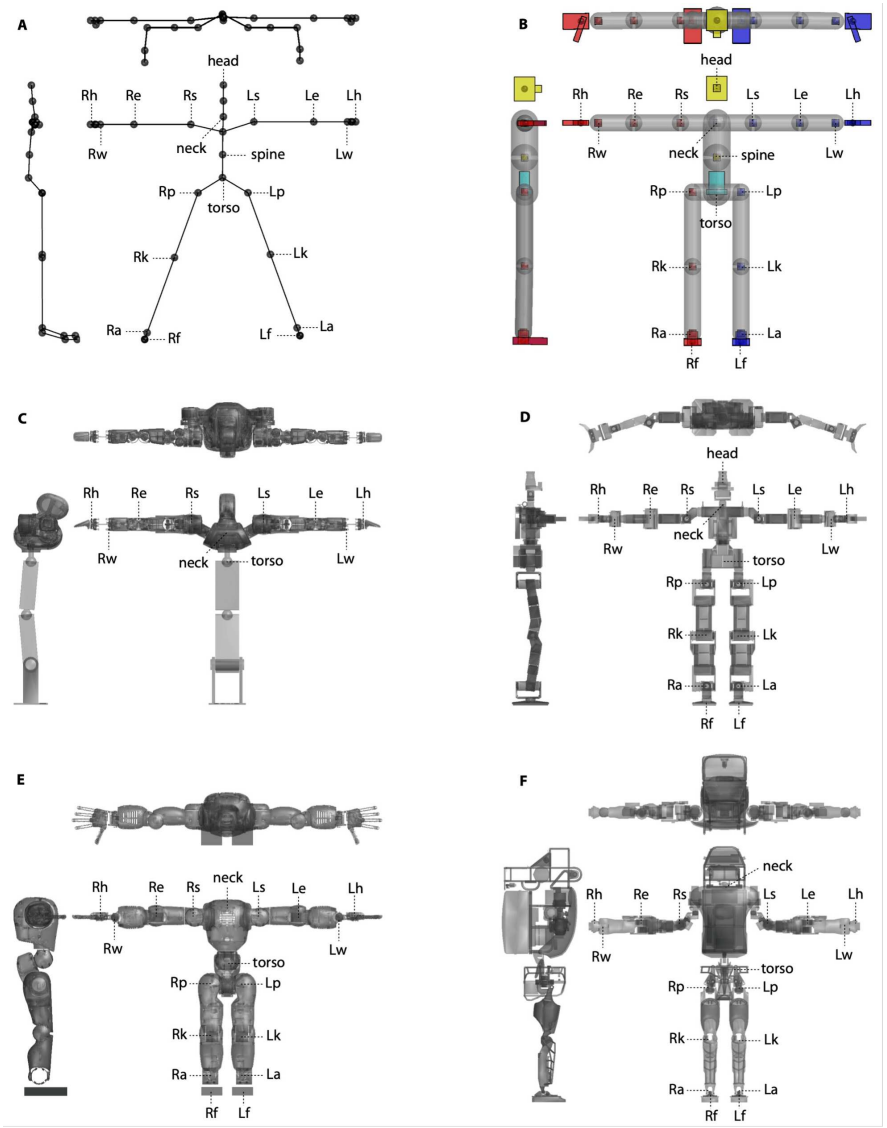

Fig. 3 Kinematic structures and JOI information. (**A**) Estimated human skeleton, which depends on the motion capture systems or the pose estimation methodologies. (**B**) Common-rig which comprises a rigid body structure along with physical attributes such as mass and moment of inertia. (**C**) AMBIDEX structure. (**D**) THORMANG structure. (**E**) COMAN structure. (**F**) ATLAS structure.

by leveraging the physical information imbued in the common rig, such as volume and mass.

2. A flexible and generalizable motion retargeting pipeline that adapts to diverse robot platforms by leveraging semantic correspondence between the common-rig and target robot joints.

3. Comprehensive experimental validation demonstrates our approach's effectiveness and practicality using diverse motion data sources and robot platforms.

2 Results

Our motion retargeting pipeline can handle both motion capture data and pose data extracted from RGB videos and transfer motions to multiple robots (Fig. 4 and Fig. 5). We utilized motion capture datasets from CMU MoCap [37] and Emotion MoCap [38], as well as pose estimation data from FrankMoCap [34] and MHFormer [36]. FrankMoCap and MHFormer were chosen for their respective strengths in whole-body motion capture and temporally consistent pose estimation. The proposed retargeting pipeline is performed on 12 different robot models, and real robot experiments are conducted on three of these models. Through our approach, robots successfully perform playful movements and even dance, as well as basic actions such as handshakes using various motion data inputs.

In evaluating the results, we focused on two main aspects: retargeting performance and robustness. Performance evaluation was based on the pipeline's ability to maintain the style of the original motion when generating robot motion. We assessed the similarity between the retargeted robot motion and the source human motion using the Procrustes Aligned Mean Per Joint Position Error (PA-MPJPE) [39–44]. The PA-MPJPE metric calculates the average Euclidean distance between corresponding joints after applying Procrustes alignment to the motion trajectories. A lower PA-MPJPE value indicates a higher similarity between the retargeted motion and the original motion. To provide a comprehensive evaluation, we computed the PA-MPJPE for a total of six joints: right shoulder, left shoulder, right elbow, left elbow, right wrist, and left wrist.

To evaluate the robustness of our approach, we conducted experiments by injecting Gaussian noise into the 3D joint positions of the ground truth human motion. We then retargeted the noisy motion to the robot using our method and the two comparison methods. The robustness was quantified by comparing the PA-MPJPE between the retargeted robot motion and the ground truth human motion. Furthermore, we demonstrated the robustness of our approach through qualitative evaluations, assessing the pipeline's ability to refine inaccurately estimated poses and maintain stable center of mass trajectories. We compared our motion retargeting method with those proposed by Ayusawa et al. [14] and Darvish et al. [17]. The performance and robustness of the three methods were evaluated on various motions from the datasets, which were retargeted to different robot models.[1]

2.1 Motion retargeting using motion capture data

We evaluated the performance and robustness of our motion retargeting method using motion capture data from the CMU MoCap and Emotion MoCap datasets and compared our approach with the methods proposed by Ayusawa et al. [14] and Darvish et al. [17]. We assessed the retargeting performance using the Procrustes Aligned

[1]This research was carried out with ethics approval from the ethics review board of Korea University under proposal KUIRB-2024-0069-01.

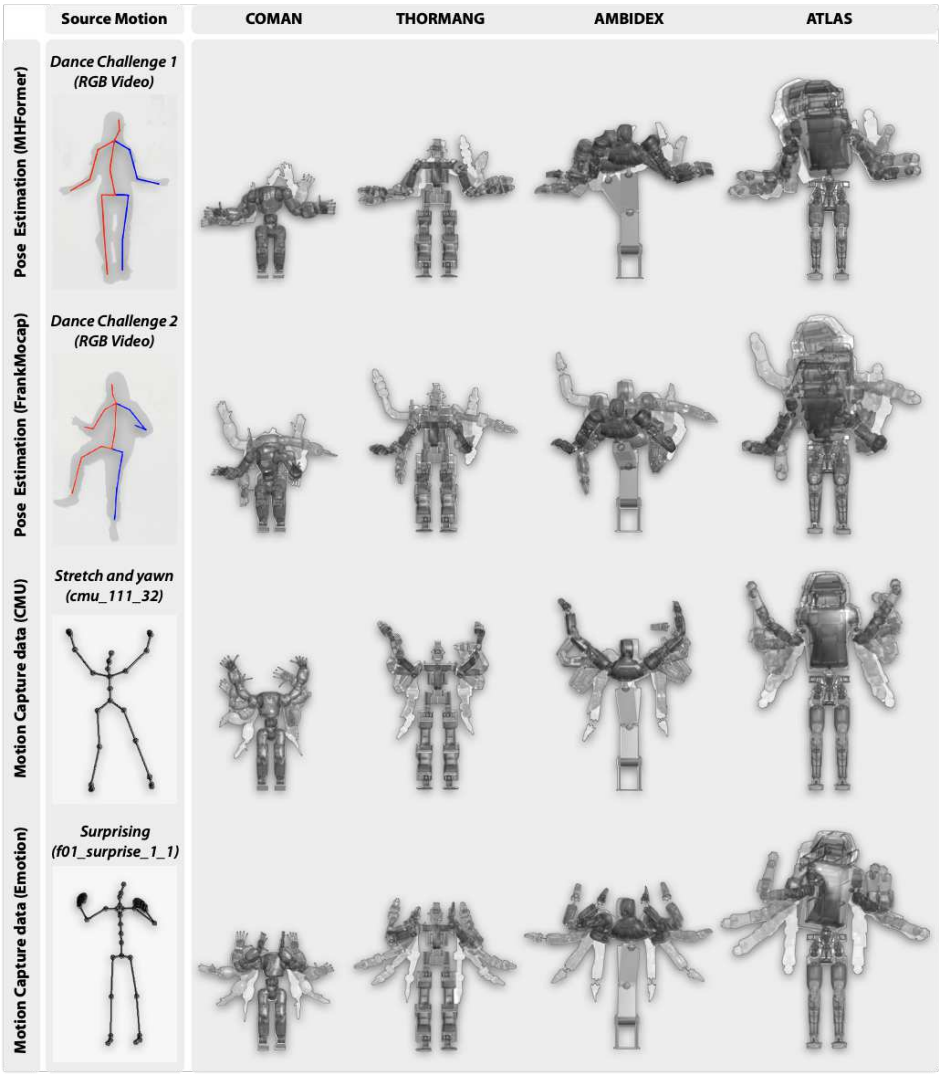

Fig. 4 Results of different robots retargeted from different types of motion data. Our motion retargeting pipeline can transfer motions to various robots using motion capture data and pose data from videos. We used different types of motion capture data, including CMU MoCap and Emotion MoCap, as well as estimated motion data from FrankMoCap and MHFormer.

Mean Per Joint Position Error (PA-MPJPE) metric, which quantifies the similarity between the retargeted robot motion and the source human motion. The PA-MPJPE is derived from the Procrustes analysis [39], which involves applying a series of linear transformations to the Cartesian trajectories $\mathbf{x}_{1:L}^{\text{rig}}$ and $\mathbf{x}_{1:L}^{\text{robot}}$, representing the original motion trajectory and the motion retargeted to a robot, respectively. First, each

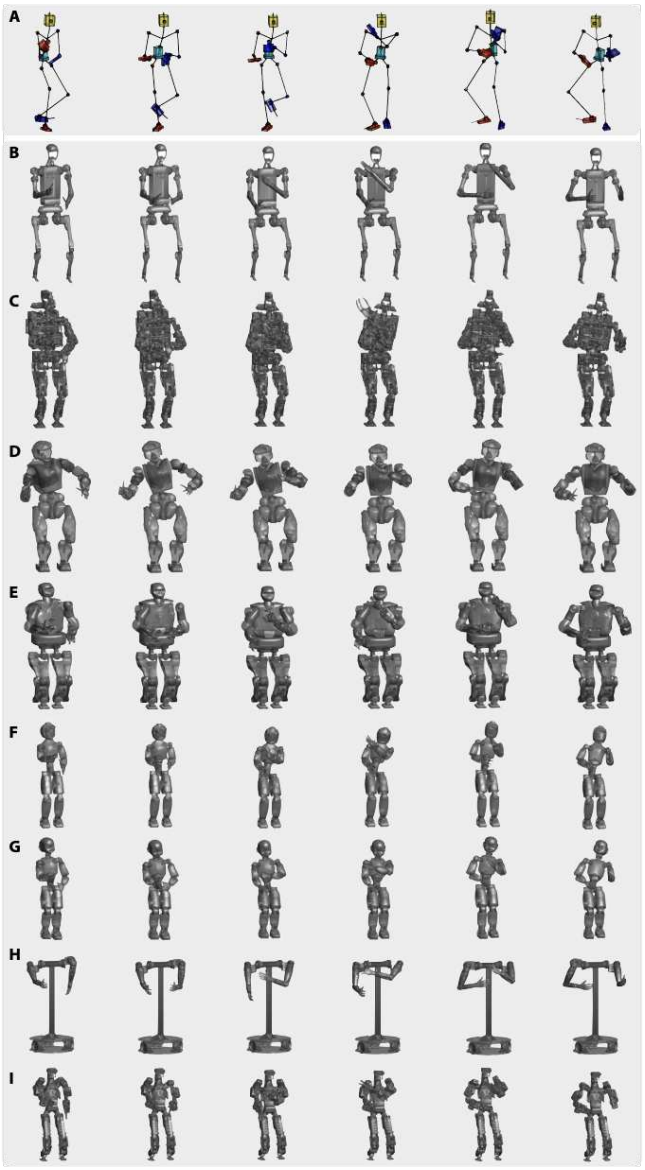

Fig. 5 Motion retargeting results for multi robots: (**A**) Source Human motion (CMU MoCap Syutouuke) (**B**) H1 (**C**) Jaxon (**D**) Valkyrie (**E**) Talos (**F**) ErgoCub (**G**) iCub (**H**) JF2 (**I**) Darco3.

curve is re-centered to its mean:

$$\hat{\mathbf{x}}^{\text{rig}}_{1:L} \leftarrow \mathbf{x}^{\text{rig}}_{1:L} - \mu(\mathbf{x}^{\text{rig}}_{1:L})$$
$$\hat{\mathbf{x}}^{\text{robot}}_{1:L} \leftarrow \mathbf{x}^{\text{robot}}_{1:L} - \mu(\mathbf{x}^{\text{robot}}_{1:L})$$

		Atlas			Coman			Thormang			AMBIDEX		
		Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish
1	cmu mocap - wash windows	**4.8556**	5.6007	10.401	**2.9434**	2.8325	12.526	**4.1490**	4.2040	5.9669	**2.3799**	2.5129	7.9924
2	cmu mocap - jumping jacks	**3.7290**	4.4364	9.1404	**3.2361**	3.7717	10.920	**3.6545**	3.9314	6.4537	**4.3740**	4.9372	8.5220
3	cmu mocap - direct traffic	**7.0674**	8.2434	10.052	**3.9889**	4.7963	11.274	**5.2467**	5.6184	9.3550	**3.9171**	4.6035	11.503
4	cmu mocap - lean forward	**4.2844**	5.8367	12.501	**5.4261**	6.4008	10.782	**5.5865**	5.6589	6.8448	**3.9842**	4.3864	6.0778
5	cmu mocap - wait for bus	**5.6924**	6.7502	6.4732	**2.9324**	3.3615	9.9185	**4.2425**	4.4020	6.5564	**4.2052**	4.7771	9.2605
6	cmu mocap - penguin	**3.5673**	3.7670	14.768	**3.4692**	3.7799	13.100	**6.7316**	6.8770	9.5163	**3.7801**	3.7885	13.886
7	cmu mocap - superhero	**2.4552**	2.8322	12.290	**4.1486**	4.3434	10.777	**6.1765**	6.3603	11.786	**3.5515**	3.8028	12.873
8	cmu mocap - lambada dance	**4.9453**	6.8977	16.166	**3.6715**	4.4453	13.504	**6.2991**	6.9555	11.087	**4.1681**	5.1701	12.819
9	cmu mocap - eating dinner	**6.9846**	7.7924	10.489	**5.9094**	6.8954	12.184	**4.9135**	5.1224	10.094	**5.8217**	6.1285	8.9128
10	cmu mocap - playing violin	**5.2599**	5.6768	8.6096	**2.8744**	3.0515	7.5437	**5.5577**	5.7297	7.6280	**3.3910**	3.5542	8.9220
11	cmu mocap - driving	**5.8861**	6.4694	9.5713	**3.2141**	3.3982	14.975	**5.6793**	5.8326	9.2542	**4.6973**	4.9180	9.0613
12	cmu mocap - stretch and yawn	**5.5028**	5.6076	6.9568	**5.0729**	5.0582	5.8363	**4.1167**	4.1873	6.0674	**5.2480**	5.3702	7.3707
13	emotion mocap - surprise	**2.8649**	3.7448	7.7360	**2.4291**	3.7383	7.1291	**1.9889**	2.7973	6.8721	**2.7105**	3.0761	9.5972
14	emotion mocap - sad	**2.1477**	2.1840	5.7173	**1.7429**	2.6510	9.3331	**1.8753**	2.8017	6.4220	**2.4009**	3.7510	7.4349
15	emotion mocap - angry1	**3.6181**	4.6965	8.8076	**2.2467**	2.9725	8.0774	**2.9761**	4.7683	7.4796	**3.2126**	4.4001	11.114
16	emotion mocap - angry2	**2.8440**	3.1530	7.6186	**1.7246**	2.4895	8.7857	**2.0627**	2.7004	6.5172	**1.6781**	4.1122	6.4867
17	emotion mocap - angry3	**3.6911**	3.9293	10.108	**2.6001**	4.2502	10.560	**2.3684**	3.1581	5.9066	**3.1176**	3.9977	10.8355
18	emotion mocap - angry4	**2.2238**	3.1757	7.4643	**2.1083**	3.1765	7.0433	**1.7725**	2.5072	5.0433	**2.3831**	3.1829	9.4434
19	emotion mocap - angry5	**2.7317**	3.0126	8.3190	**2.0571**	2.8116	7.5696	**2.7006**	3.1295	6.3948	**2.3463**	2.6917	9.2652
20	emotion mocap - disgust1	**2.3677**	3.0319	8.0295	**2.5812**	2.7683	9.3834	**1.8532**	2.3365	6.7240	**2.4098**	2.9606	9.4186
21	emotion mocap - disgust2	**2.8140**	3.2223	8.7626	**2.6434**	3.0907	8.9569	**2.7945**	3.9486	8.1071	**2.7058**	4.0184	10.738
22	emotion mocap - fearful1	**2.7613**	2.7880	9.6380	**2.2345**	2.8138	8.2061	**2.3162**	2.7086	7.0222	**1.8019**	2.2881	9.0820
23	emotion mocap - fearful2	**2.0234**	2.4694	6.9216	**1.5288**	2.4998	6.2864	**1.2296**	1.8440	5.9935	**1.7188**	2.4228	7.8657
24	emotion mocap - happy	**2.7180**	2.9507	6.8090	**1.9667**	3.2792	7.5843	**2.7687**	3.1886	5.8459	**2.6157**	2.9049	5.8146

Table 1 Performance evaluation of our motion retargeting approach compared to two motion retargeting methods. The Procrustes aligned mean per joint position error (PA-MPJPE) between the retargeted robot motion and the ground truth human motion is reported for CMU, Emotion MoCap data. Our method achieves the lowest PA-MPJPE score for all motions and robots.

where $\mu(\cdot)$ is a column-wise mean operator. Then, the re-centered curves are scaled with respect to the Frobenius norm $||\cdot||_F$:

$$\tilde{\mathbf{x}}^{\text{rig}}_{1:L} \leftarrow \frac{\hat{\mathbf{x}}^{\text{rig}}_{1:L}}{||\hat{\mathbf{x}}^{\text{rig}}_{1:L}||_F}$$

$$\tilde{\mathbf{x}}^{\text{robot}}_{1:L} \leftarrow \frac{\hat{\mathbf{x}}^{\text{robot}}_{1:L}}{||\hat{\mathbf{x}}^{\text{rig}}_{1:L}||_F}$$

The optimal rotation matrix $R \in \text{SO}(3)$ to align the trajectories is computed by:

$$R^* = \arg\min_R ||R\tilde{\mathbf{x}}^{\text{robot}}_{1:L} - \tilde{\mathbf{x}}^{\text{rig}}_{1:L}||_F$$

which can be solved via Singular Value Decomposition. Finally, the PA-MPJPE is defined as:

$$d(\mathbf{x}^{\text{rig}}_{1:L}, \mathbf{x}^{\text{robot}}_{1:L}) = ||R^*\tilde{\mathbf{x}}^{\text{robot}}_{1:L} - \tilde{\mathbf{x}}^{\text{rig}}_{1:L}||_2$$

which is the sum of Euclidean distances between corresponding joints after the Procrustes alignment.

Table 1 presents the PA-MPJPE scores for 24 different motions, retargeted to four robot models. Our method consistently achieves the lowest PA-MPJPE values compared to the other two approaches, demonstrating its superior ability to preserve the style and accuracy of the original motion. To evaluate the robustness of our approach in handling noisy motion data, we injected Gaussian noise with a standard deviation of 0.02 to the 3D joint positions of the ground truth human motion and then retargeted the noisy motion to the robot using the three methods. Table 2 presents the PA-MPJPE scores for 12 different motions retargeted to four robot models under noisy conditions. Our method achieves the lowest PA-MPJPE scores for the majority of the motions and robot models, with only a few exceptions where Ayusawa et al.'s method

		Atlas			Coman			Thormang			AMBIDEX		
		Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish	Ours	Ayusawa	Darvish
1	cmu mocap - unscrew bottlecap	**0.1465**	0.2662	0.4301	**0.1491**	0.2337	0.4856	**0.2096**	0.2342	0.2234	**0.1227**	0.2499	0.3624
2	cmu mocap - chicken dance	**0.0639**	0.1259	0.5517	**0.1098**	0.1512	0.6026	**0.1101**	0.1385	0.2666	**0.3454**	0.5545	0.5181
3	cmu mocap - bear	**0.0572**	0.0898	0.2789	**0.0591**	0.1049	0.5629	**0.0903**	0.2233	0.6231	**0.1207**	0.1771	0.3611
4	cmu mocap - crying	**0.2084**	0.4094	0.5779	**0.4422**	0.5085	0.6547	**0.3480**	0.4967	0.5626	**0.3246**	0.5722	0.4638
5	cmu mocap - scared	**0.2395**	0.2496	0.6481	0.2651	**0.2531**	0.6973	**0.2350**	0.2402	0.6031	**0.4108**	0.4631	0.5371
6	cmu mocap - upset	**0.1158**	0.1515	0.3761	0.1124	**0.0963**	0.3210	**0.1126**	0.2009	0.1438	**0.1645**	0.1695	0.3592
7	cmu mocap - chugging	**0.1721**	0.3027	0.3618	**0.2013**	0.2283	0.3964	**0.1926**	0.2129	0.3670	**0.1839**	0.2372	0.2718
8	cmu mocap - making coffee	**0.1562**	0.2309	0.5227	**0.1503**	0.1737	0.5628	**0.1341**	0.1600	0.2224	**0.1515**	0.2075	0.5419
9	cmu mocap - violin	**0.0772**	0.1312	0.5664	**0.0916**	0.1207	0.3097	0.1196	**0.1094**	0.2534	**0.0668**	0.1075	0.4362
10	cmu mocap - waving	**0.0502**	0.1079	0.3464	**0.0581**	0.0773	0.3553	**0.0505**	0.0790	0.2749	**0.0482**	0.0890	0.3983
11	cmu mocap - macarena dance	**0.1235**	0.1386	0.4884	0.0956	**0.0825**	0.3588	0.0934	**0.0841**	0.2496	**0.0999**	0.1183	0.8012
12	cmu mocap - sun salutation	**0.0452**	0.0470	0.2689	**0.0658**	0.0722	0.2586	**0.0395**	0.2409	0.2053	**0.0399**	0.2573	0.3617

Table 2 Robustness evaluation of our motion refinement approach compared to two motion retargeting methods. The Procrustes aligned mean per joint position error (PA-MPJPE) between the retargeted robot motion and the ground truth human motion is reported for 12 different motions. Our method generally achieves low PA-MPJPE scores.

slightly outperforms ours. These results highlight the effectiveness of our approach in refining noisy motion data and generating feasible and accurate robot motions, even in the presence of noise.

In addition to the four robot models presented in Table 1 and Table 2, we applied our method to eight more robot models in simulation. Fig. 5 showcases the retargeted motions for these additional robot platforms, demonstrating the versatility and effectiveness of our approach in generating visually appealing and human-like motions across a wide range of robot morphologies. Furthermore, we validated the practicality and effectiveness of our method on three real robot platforms: AMBIDEX, Thormang, and JF2. Fig. 6 presents snapshots of the retargeted motions executed on these robots. The successful execution of the generated motions on real robot hardware highlights the potential of our approach for real-world applications, as it demonstrates the ability to generate smooth, stable, and expressive robot motions that closely resemble the original human motions.

2.2 Motion retargeting with RGB videos

The proposed motion retargeting pipeline was designed to handle motion data estimated from RGB videos, as well as MoCap data. However, the estimated human motion data is generally less accurate compared to data acquired from motion capture devices. When extracting motion data from 2D images, depth information may be lost or ambiguous, which can negatively affect the accuracy of 3D motion estimation. Consequently, this problem can result in noisy motion data. Furthermore, the extracted human motion data from pose estimation may exhibit variations in skeletal structures, depending on the algorithm used and the specific key points being tracked.

To retarget human motion extracted from pose estimation to robots, it is necessary to convert a non-fixed skeleton to a fixed-link skeleton and refine noisy motion. Furthermore, the motion needs to be adjusted by considering physical constraints such as the robots' kinematic structure, velocity-acceleration limits, and collision avoidance. Trajectory smoothing is also required to eliminate the motion jitter that occurs during the estimation process. Our robot motion retargeting pipeline addresses these challenges and generates feasible motions for multiple robots.

We collected diverse motion data from human motion videos using two pose estimation methods: FrankMoCap [34], which can obtain wrist orientation information

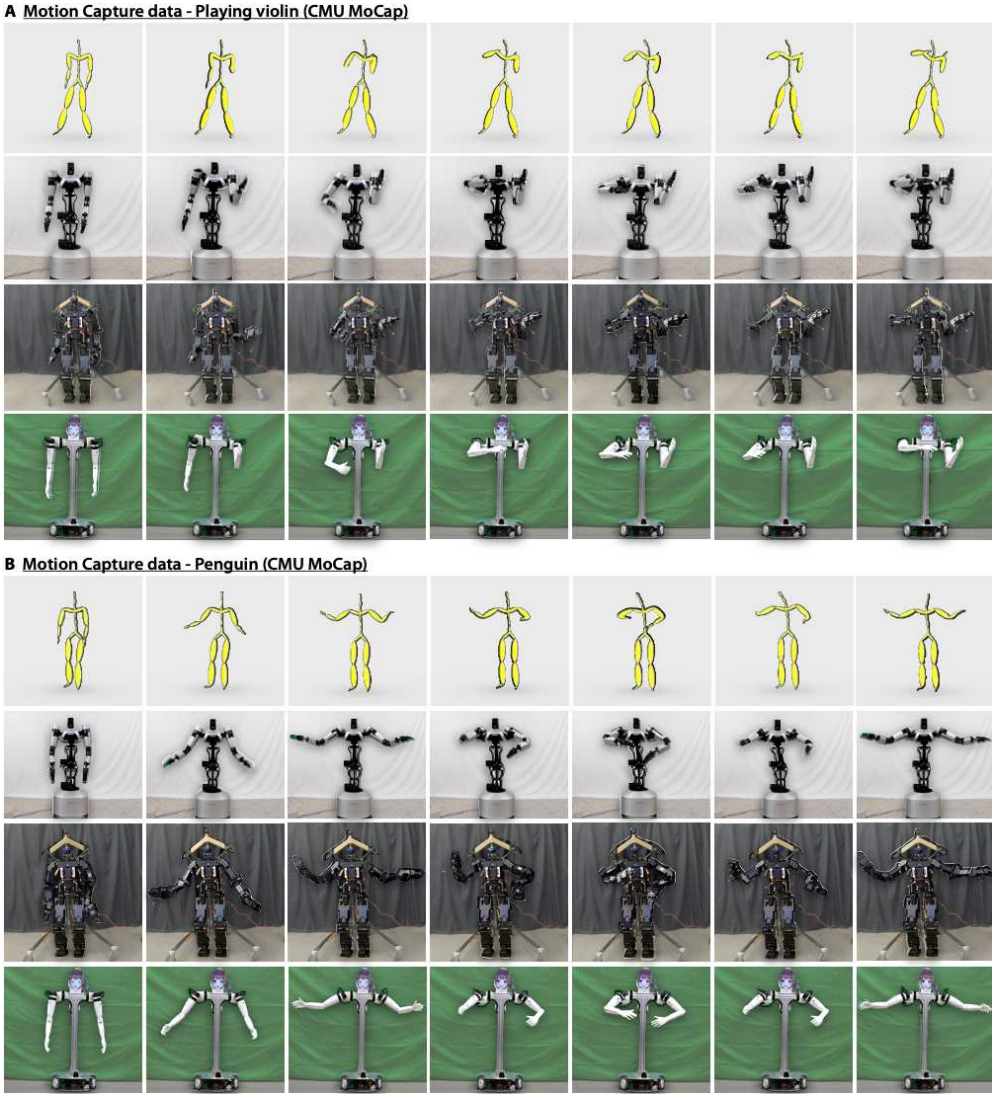

Fig. 6 Results of retargeting from Motion capture data to robots. (**A**) From top to bottom, human motion on for play violins in CMU-MoCap retargeted AMBIDEX, THORMANG, and JF2's motion. (**B**) From top to bottom, human motion on for Penguin in CMU-MoCap retargeted AMBIDEX, THORMANG, and JF2's motion.

but sometimes produces unsmooth motion, and MHFormer [36], which can address occlusion and generate smooth motion without considering the wrist joint. The motion dataset was retargeted to four robots. We first demonstrated the robustness of our methodology in handling noisy motions, such as tilting the torso to one side, standing on tiptoes, or having the feet off the ground. We compared the whole-body posture of the motion with and without post-rigging applied at the same time frame. The results

in Fig. 7A clearly show that the noisy source motion (left) is refined into a post-rigging motion (right) with an upright posture and both feet firmly attached to the ground, providing stable support for the body.

Secondly, we checked the extent to which the trajectory of the center of mass, projected onto the ground, goes beyond the convex hull formed between the two feet. If the center of mass goes beyond the convex hull, we determined that the motion may cause the robot to fall and, thus, consider it unstable. Fig. 7B shows the result of plotting the projected CoM (Center of Mass) trajectory and convex hull together. As illustrated in Fig. 7B, the CoM trajectory of the motion with post-rigging was formed within the convex hull, while the CoM trajectory of the motion without post-rigging frequently went beyond the designated area.

In Fig. 8A, snapshots of RGB video, AMBIDEX, and JF2 motions for the same motion are shown. Despite the difference in movement between people in the RGB videos and the stationary robots, we successfully retargeted robots by considering the characteristics of the original motion. Furthermore, we have also demonstrated the applicability of our methodology to the THORMANG robot, which is not fixed to the ground and relies on two feet for support. Fig. 8B represents the result of retargeting the motion captured from an RGB video recorded with an iPhone13 to robots.

3 Discussion

3.1 Rig unification and reusable motion data

Previous studies have required the analysis of diverse skeleton structures present in motion data and subsequent manual transfer to robots. However, the process of manually generating robot motion is time-consuming and offers limited reusability when modifications are made to joint configurations. In this study, we propose a pre-rigging step in the common-rigging process, which allows for the unification of various skeleton rigs into a single predefined rig known as the common-rig. The common-rig's motion is easily transferable to different types of robots, owing to its composition of revolute joints and simple structure. The motion becomes reusable and can be effortlessly transformed to generate a variety of humanoid robots with minimal manual labor. The common-rig representation serves as a foundation for our motion retargeting pipeline, enabling us to handle diverse motion sources and generate high-quality robot motions. By abstracting away the complexities of the source data and providing a consistent representation, common-rigging simplifies the retargeting process and enhances its robustness. Additionally, the motion associated with the common-rig can serve as a valuable database for future research in learning-based motion retargeting, thereby providing opportunities for advancements in diverse approaches to robot motion retargeting.

3.2 Refinement of noisy motions

Previous studies [15, 20] have highlighted the challenge of transferring motion data obtained from RGB videos to robots due to the inaccuracies in pose estimation caused by the lack of depth information. In our methodology, the common-rigging process

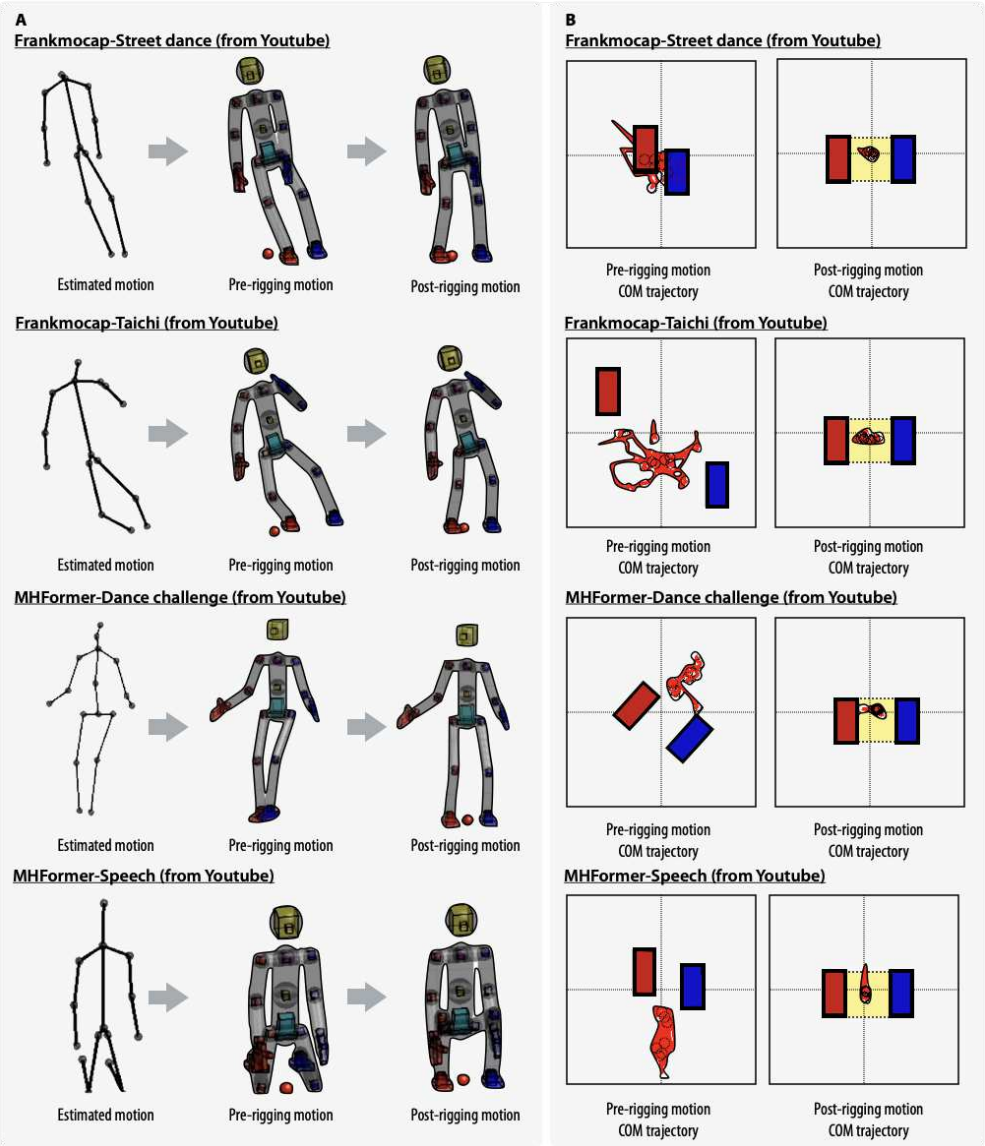

Fig. 7 Refinement of noise motions. (**A**) The inaccurately estimated original motion (left) is corrected to an upright posture after going through our methodology, and both feet are attached to the ground, enabling the body to be supported (right). (**B**) The result of plotting the projected CoM trajectory and convex hull together. the CoM trajectory of the motion with post-rigging is formed within the convex hull, while the CoM trajectory of the motion without post-rigging frequently goes beyond the designated area.

effectively refines noisy motion estimated from pose estimation into feasible motion by considering the common-rig's physical properties. The common-rig representation

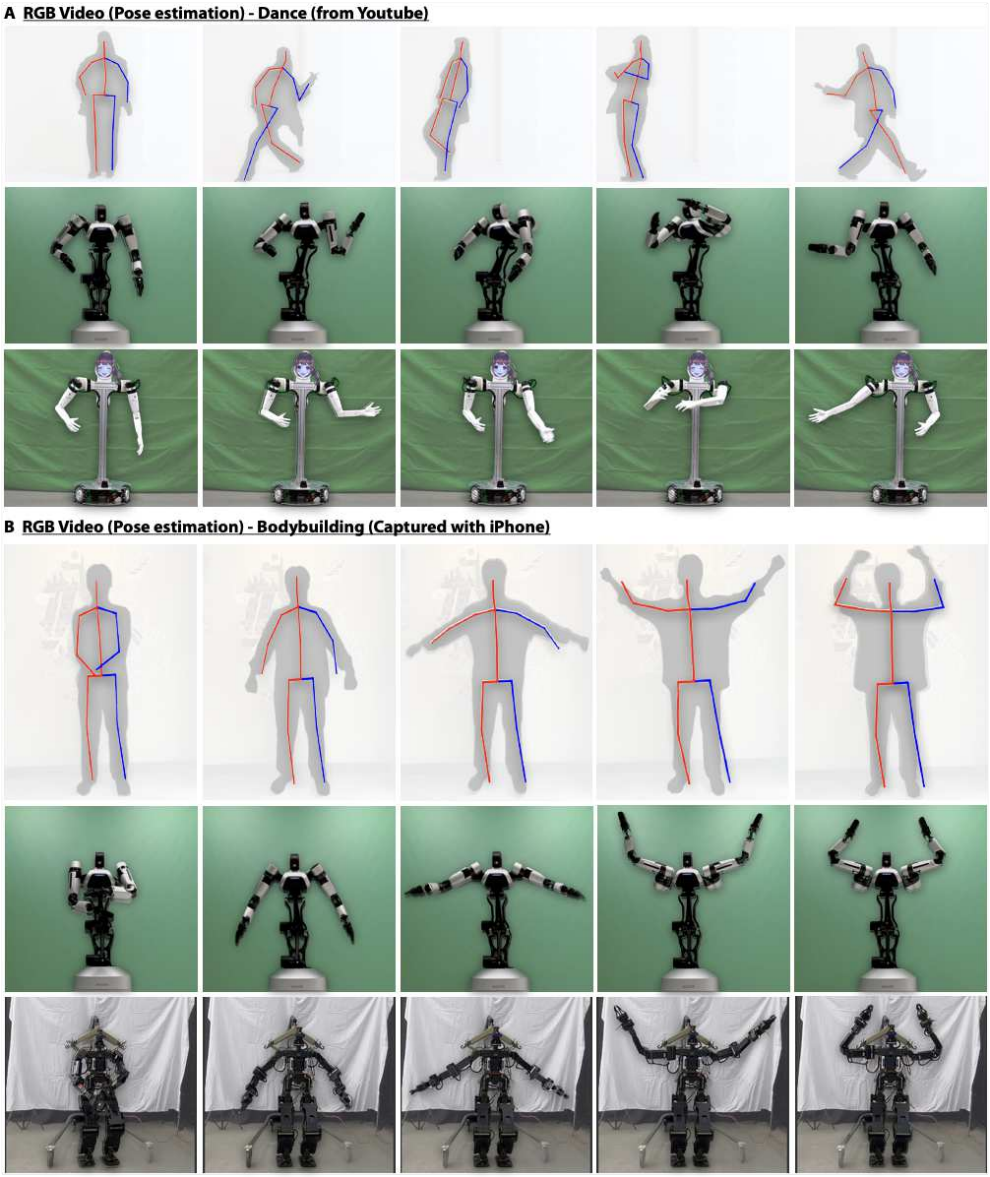

Fig. 8 Result of retargeting from estimated motion data from RGB video to robots. (**A**) The result of retargeting from the moving person's image to the robot with the base fixed. By considering the original motion's characteristics, we successfully retargeted AMBIDEX and JF2. (**B**) The result of retargeting the motion obtained from the RGB video taken with the iPhone to AMBIDEX and THORMANG.

incorporates physical information such as link lengths, mass, and volume, which are essential for generating physically plausible and stable motions for the target robot.

The post-rigging optimization enforces ground contact preservation, motion smoothing constraints, and self-collision avoidance, effectively improving the quality and feasibility of the motions before transferring them to the robot. Through experiments, we have validated the effectiveness of our methodology in refining inaccurate poses, such as tilted poses.

3.3 Easy-to-use motion retargeting pipeline for multiple robots

Our method effortlessly retargets human motions to multiple robot motions with just a few simple steps. The only manual requirement in our methodology is the initial setup of the JOI for the specific robot being retargeted. By utilizing our method, individuals can easily retarget their captured videos, such as dance challenges, to robots. In essence, our motion videos can be transformed into robot motions at any given time. This will enable robots to perform a wide range of motions, enhancing human interaction and creating long-lasting magical experiences. We successfully retargeted diverse motion data acquired from motion capture and RGB video sources onto four simulated robots. Furthermore, through application to two real robots, we provided evidence of the effectiveness of our methodology in real-world environments.

3.4 Limitations and Future Work

Performing motions with a robot when its feet are not in contact with the ground is very challenging as it requires considering various factors such as contact and balancing [45]. Our methodology focuses on refining motion, assuming that the robot's two feet remain in contact with the ground, and applying it to the robot using open-loop control. For this reason, THORMANG was operated at a slightly slower speed for dynamic motions that carried a risk of falling. However, the limitations can be adequately addressed by utilizing the whole-body quadratic program controller proposed in previous studies [19, 46, 47].

We have demonstrated the superior performance of our methodology compared to other approaches by calculating the Procrustean distance in the 3D Cartesian space between the source motion and the robot motion. While this evaluation method assesses how well the robot motion trajectory preserves the form of the source motion trajectory, it has limitations in evaluating the naturalness of the robot. Expressing naturalness numerically is challenging as it is subjective and varies from person to person [3]. However, we have observed that completely imitating a person may not always result in natural robot motion; instead, the relationship between the hardware's weight and motion speed can influence naturalness. Therefore, as a future study, we will further investigate the naturalness of robot motion based on this study's generated robot motion data.

Our current implementation may not be real-time due to the computational cost of solving complex non-convex optimization problems, particularly in the pre-rigging, post-rigging, and motion retargeting steps. We are actively working on improving the speed of our pipeline through code optimization and the integration of learning-based and sampling-based motion retargeting techniques [28, 29]. The ongoing work on porting the code to C/C++ and utilizing the Mujoco simulator has already shown

promising results in reducing the computation time. We believe that these efforts, along with the incorporation of learning and sampling-based approaches, will significantly enhance the computational efficiency of our approach, bringing us closer to real-time motion retargeting while ensuring the feasibility and safety of the generated motions.

4 Methods

4.1 Problem formulation

In this section, we present the results of our robust motion retargeting pipeline, which is designed to generate feasible robot motions across various kinematic configurations. We evaluate the performance of our method using different types of source motion data, including motion capture data and RGB videos. The experiments demonstrate the effectiveness of our approach in handling noisy and inconsistent input data, as well as in generating stable and collision-free robot motions that adhere to the robot's kinematic constraints.

4.1.1 Notations

The main notations used throughout the paper are as follows:

- $\xi^{\mathcal{K}}$: The kinematic structure of a skeleton, where $\mathcal{K} \in \{\text{mocap}, \text{rig}, \text{robot}\}$. ξ^{mocap}, ξ^{rig}, and ξ^{robot} represent the kinematic structures of a motion capture skeleton, a common-rig, and the target robot, respectively.
- $\mathrm{T}^{\mathcal{K}}_{1:L} \in \mathrm{SE}(3)^{L \times N}$: The homogeneous transformation matrices of skeleton joints in the motion sequence. The notation $1:L$ refers to the interval from the start of the motion to the L-th tick, and N indicates the number of skeleton joints.
- $q^{\mathcal{K}}_{1:L} \in \mathbb{R}^{L \times N}$: The joint angles of the skeleton in the motion sequence. L is the length of the input motion sequence, and N indicates the number of skeleton joints.
- $\mathrm{FK}\left(\xi^{\mathcal{K}}, q^{\mathcal{K}}_{1:L}\right) \mapsto \mathrm{T}^{\mathcal{K}}_{1:L}$: The forward kinematics function that outputs homogeneous transformation matrices, given $\xi^{\mathcal{K}}$ and $q^{\mathcal{K}}_{1:L}$ as inputs.

4.1.2 Motion retargeting formulation

Our motion retargeting formulation is as follows:

$$
\begin{aligned}
\min_{q^{\text{robot}}_{1:L}} \quad & d\left(\hat{\mathrm{T}}^{\text{robot}}_{1:L}, \mathrm{FK}\left(\xi^{\text{robot}}, q^{\text{robot}}_{1:L}\right)\right) \\
\text{s.t.} \quad & q^{\text{rig}}_{1:L} = \mathrm{CR}\left(\xi^{\text{rig}}, \xi^{\text{mocap}}, \mathrm{T}^{\text{mocap}}_{1:L}\right) && \text{(1a)} \\
& \mathrm{T}^{\text{robot}}_{1:L} = \mathrm{RT}\left(\xi^{\text{robot}}, \xi^{\text{rig}}, q^{\text{rig}}_{1:L}\right) && \text{(1b)} \\
& \hat{\mathrm{T}}^{\text{robot}}_{1:L} = \mathrm{FT}\left(\mathrm{T}^{\text{robot}}_{1:L}, \mathrm{T}^{\text{mocap}}_{1:L}\right) && \text{(1c)} \\
& g\left(\xi^{\text{robot}}, q^{\text{robot}}_{1:L}\right) \geq 0 && \text{(1d)}
\end{aligned}
$$

In this formulation, we first convert the source motion into feasible robot motion, where the common-rigging (Eq. 1a) and the direction vector-based robot target

pose (Eq. 1b) correspond to this step. The common-rigging is defined as CR : $\left(\xi^{\text{rig}}, \xi^{\text{mocap}}, \mathrm{T}_{1:L}^{\text{mocap}}\right) \mapsto q_{1:L}^{\text{rig}}$. CR finds the sequence of corresponding common-rig configurations, $q_{1:L}^{\text{rig}}$, given the motion capture data, ξ^{mocap}, and $\mathrm{T}_{1:L}^{\text{mocap}}$. The common-rig, ξ^{rig}, is a pre-made auxiliary rig in the form of a human. We divide this process into two steps: pre-rigging and post-rigging. The direction vector-based robot target pose is defined as RT : $\left(\xi^{\text{robot}}, \xi^{\text{rig}}, q_{1:L}^{\text{rig}}\right) \mapsto \mathrm{T}_{1:L}^{\text{robot}}$. RT derives the target pose of the robot, $\mathrm{T}_{1:L}^{\text{robot}}$, by combining the link length information of ξ^{robot} and the directional information of ξ^{rig}. The directional information is derived from $\mathrm{T}_{1:L}^{\text{rig}}$, which is the output of FK $\left(\xi^{\text{rig}}, q_{1:L}^{\text{rig}}\right)$.

Finally, we perform the robot motion trajectory adjustment (Eq. 1c) to preserve the similarity between the robot motion and the original motion in the task space. The process is not applied to all joints, but only to the set of hand and elbow joints. It is defined as FT : $\left(\mathrm{T}_{1:L}^{\text{robot}}, \mathrm{T}_{1:L}^{\text{mocap}}\right) \mapsto \hat{\mathrm{T}}_{1:L}^{\text{robot}}$. FT derives $\hat{\mathrm{T}}_{1:L}^{\text{robot}}$, which represents the final goal for the robot, by minimizing the trajectory error between $\mathrm{T}_{1:L}^{\text{robot}}$ and $\mathrm{T}_{1:L}^{\text{mocap}}$ in the task space. $\hat{\mathrm{T}}_{1:L}^{\text{robot}}$ becomes the desired final position for the robot to reach. To obtain the robot's joint angles $q_{1:L}^{\text{robot}}$, we solve a numerical inverse kinematics problem [48] that minimizes the Euclidean distance d between the target pose $\hat{\mathrm{T}}_{1:L}^{\text{robot}}$ and the robot's pose FK $\left(\xi^{\text{robot}}, q_{1:L}^{\text{robot}}\right)$ obtained by forward kinematics.

The inequality constraint $g(\cdot)$ in (Eq. 1d) ensures the smoothness and feasibility of the generated robot motion by addressing three main aspects: self-collision avoidance, the center of mass (COM) constraint, and motion smoothing. To prevent self-collisions, collision boundaries are defined based on the capsule approximations of the robot's body parts. The inequality constraint maintains the motion within these boundaries, avoiding collisions between different parts of the robot. Similarly, the COM constraint ensures that the projected COM remains within the support area formed by the robot's feet, maintaining the robot's stability.

Motion smoothing is achieved by considering the robot's joint velocity and acceleration limits. We employ the Gaussian Random Path (GRP) method [49], a Gaussian process-based approach, to generate smooth trajectories that satisfy these limits. The GRP algorithm optimizes the motion trajectory while keeping the velocities and accelerations within the specified bounds. By incorporating these limits into the inequality constraint $g(\cdot)$, we ensure that the resulting motion is smooth and free from abrupt changes or jerky movements. The combination of self-collision avoidance, COM constraint, and motion smoothing through velocity and acceleration limit constraints in $g(\cdot)$ guarantees that the generated robot motion is both smooth and physically feasible, respecting the robot's dynamic limitations and maintaining its stability throughout the execution of the retargeted motion.

The proposed motion retargeting pipeline consists of common-rigging and robot motion retargeting processes. The common-rigging process consists of two steps: Pre-rigging and Post-rigging. Through the common-rigging process, the diverse skeletons of motion are unified into a single predefined rig called common-rig. These motions are then refined to become feasible by the robot, considering factors such as the center of mass (COM) and ground contact. The robot motion retargeting process consists of three steps: Direction vector-based robot target pose, Robot motion trajectory

adjustment, and Post-processing. The common-rig's motions are retargeted to multiple robots through the robot motion retargeting process. The target trajectory in the robot's task space is computed using the direction vector-based approach by considering the different link lengths between the human and the robot. The target trajectory then serves as the input for the inverse kinematics (IK) problem. We numerically solve the IK problem to obtain the robot's joint angles that match the target trajectory while satisfying physical constraints, such as joint velocity-acceleration limits and self-collision avoidance.

4.2 Common-rigging

The common-rigging process is designed to address two main challenges in motion retargeting: the diversity of skeleton structures in motion data and the presence of noise or physically implausible poses in estimated motions. Motion data can be obtained from various sources, such as motion capture systems, or estimated from RGB videos using pose estimation methods. However, the skeleton structures may vary depending on the motion capture devices or pose estimation algorithms used, making it difficult to directly apply the motion data to the target robot. Additionally, motion data obtained from RGB videos often contains noise or physically implausible poses due to the lack of depth information and the limitations of current pose estimation methods in handling occlusions and complex poses. To tackle these issues, we introduce a two-step common-rigging process: pre-rigging and post-rigging.

In the pre-rigging step, we aim to unify the diverse skeleton structures into a single, standardized representation called the common-rig. The common-rig is designed to have a unified kinematic structure with consistent link lengths, as well as physical properties such as volume and mass, which are computed based on the average male body size and density. To retarget the motion data onto the common-rig, we first define a set of joints of interest (JOI) following the approach proposed by [50], which typically include the head, shoulders, elbows, wrists, pelvis, knees, and ankles (Fig. 3). Then, for each pose in the motion data, we solve an inverse kinematics problem to determine the common-rig's pose that matches the positions and orientations of the JOI. The post-rigging step focuses on refining the retargeted motions to ensure their feasibility for robot execution. This is achieved by applying three main constraints simultaneously: maintaining the feet on the ground while keeping the center of mass within the support polygon, avoiding self-collisions, and smoothing the joint trajectories. By incorporating these constraints, the post-rigging step enhances the mapped motions, making them more suitable for robot motion retargeting.

The common-rigging process offers several key advantages in addressing the challenges of motion retargeting. First, by imbuing the common-rig with physical properties such as volume and mass, we enable the detection and handling of self-collisions that may arise from noisy pose estimation results. Second, the incorporation of the whole-body center-of-mass information allows for the correction of physically implausible poses, such as excessive leaning or imbalance. Finally, the use of a single, standardized kinematic structure facilitates the unification of diverse motion data from various sources, such as motion capture devices and RGB videos. To achieve these objectives, we formulate the common-rigging process as a motion trajectory

optimization problem:

$$\min_{q^{\text{rig}}_{1:L}} \quad d\left(\mathrm{T}^{\text{mocap}}_{1:L}, \mathrm{FK}\left(\xi^{\text{rig}}, q^{\text{rig}}_{1:L}\right)\right)$$

$$\text{s.t.} \quad \mathtt{LL}(\xi^{\text{rig}}, i) = \mathtt{LL}(\xi^{\text{mocap}}, i), \quad i = 1 \cdots N \tag{2a}$$

$$h(\xi^{\text{rig}}, q^{\text{rig}}_{1:L}) = 0 \tag{2b}$$

$$g(\xi^{\text{rig}}, q^{\text{rig}}_{1:L}) \geq 0 \tag{2c}$$

The objective of the optimization is to find the sequence of common-rig joint angles, $q^{\text{rig}}_{1:L}$, that minimizes the Euclidean distance d between the source motion trajectory $\mathrm{T}^{\text{mocap}}_{1:L}$ and the common-rig motion trajectory $\mathrm{FK}\left(\xi^{\text{rig}}, q^{\text{rig}}_{1:L}\right)$ in Cartesian space. The three constraints in the optimization problem, represented by $\mathtt{LL}(\cdot)$, $h(\cdot)$, and $g(\cdot)$, serve different purposes.

$\mathtt{LL}(\cdot)$ in constraint (Eq. 2a) represents the link length constraint, which ensures that the link lengths of the common-rig (ξ^{rig}) match the link lengths of the source motion skeleton (ξ^{mocap}). This constraint maintains the proportions of the original motion when converting it to the common-rig representation. $h(\cdot)$ in constraint (Eq. 2b) is an equality constraint that keeps the feet of the common-rig in contact with the ground throughout the motion, preventing the feet from penetrating the ground or floating in the air. This constraint helps generate stable and physically plausible motions.

The inequality constraint $g(\cdot)$ in (Eq. 2c) ensures the smoothness and feasibility of the generated robot motion by addressing three main aspects: motion smoothing, self-collision avoidance, and maintaining the robot's center of mass within the support polygon. Motion smoothing is achieved by imposing limits on the joint velocities and accelerations of the common-rig, reducing abrupt changes or jerky movements in the resulting motion. Self-collision avoidance is enforced by constraining the signed distance between the meshes of different body parts to be positive, preventing any intersections during the motion. The center of mass constraint ensures that the robot maintains its balance by keeping the center of mass within the contact polygon defined by the robot's feet. By incorporating these aspects into the inequality constraint, the generated robot motion becomes smooth, collision-free, and dynamically stable, respecting the robot's physical limitations.

By solving this optimization problem, our algorithm generates a refined motion trajectory that significantly improves upon the input data in terms of physical plausibility and stability while maintaining the desired motion style.

4.2.1 Pre-rigging

During the pre-rigging step, motion data from various human skeletons are retargeted to the motion of a common-rig. The link lengths of the common-rig are determined based on the type of input motion data. When using motion data from pose estimation methods, we calculate the link lengths from the initial pose of the estimated motion, ensuring that the common-rig's proportions match the specific individual in the video. To construct the common-rig, we directly adopt the skeleton's link length information

provided in the mocap data for motion capture data. This process is related to the link length constraint $\texttt{LL}(\cdot)$ in (Eq. 2a), which ensures that the link lengths of the common-rig match the link lengths of the source motion skeleton. By enforcing this constraint, we maintain the proportions of the original motion when converting it to the common-rig representation.

To retarget the motion data to the common-rig, we solve an inverse kinematics problem by matching the task space positions and rotations of the JOI in the source motion data to the corresponding joints in the common-rig. This process allows us to obtain a motion that is adapted to the common-rig's skeleton while preserving the original motion's characteristics. By retargeting the motion data to the common-rig, we imbue the motion with physical information such as volume, mass, and moment of inertia. This physical information is essential for detecting and handling self-collisions, correcting physically implausible poses, and maintaining the robot's balance during the post-rigging step.

Additionally, to prevent the robot from falling due to unnatural poses, target orientations of the ankle joints are adjusted during the pre-rigging step, ensuring that the sole of the foot remains parallel to the ground. The adjusted rotation targets of ankles are obtained by solving the following optimization problem:

$$\min_{\hat{R}^{\text{rig}}_{i\in ankle}} \left\| \hat{R}^{\text{rig}}_{i\in ankle} \mathbf{e}^{\text{rig}}_{\text{z},i\in ankle} - \mathbf{e}_{\text{z}} \right\|_2 \tag{3}$$

Let $\hat{R}^{\text{rig}}_{i\in ankle} \in \text{SO}(3)$ denote the adjusted rotation targets for the ankle joint, $\mathbf{e}^{\text{rig}}_{\text{z},i\in ankle}$ denote the z-axis component of the relative coordinate basis of the common-rig ankle joints, and $\mathbf{e}_{\text{z}} = [0\ 0\ 1]^T$ denote the standard basis aligned towards the z-axis on the global coordinate system. By minimizing the cost function (Eq. 3), we can obtain the adjusted rotation targets that align the foot's sole with the ground surface.

The pre-rigging step ensures that the motion data is compatible with the common-rig and incorporates the necessary physical information for further processing in the post-rigging step. By retargeting the motion to the common-rig and adjusting the ankle orientations, we create a foundation for generating stable and physically plausible motions for the target robot.

4.2.2 Post-rigging

When a robot performs motions without its feet in contact with the ground, it becomes highly challenging due to the necessity of considering various factors such as contact and balance [45]. If human motions are directly applied to a robot under such situations, the robot is likely to fall over. To address these challenges, a post-rigging step modifies the lower-body motion of the common-rig. However, it is important to note that modifying the lower-body motion does not mean completely disregarding it. Instead, we aim to create physically plausible motions that still capture the characteristics of the original whole-body motion. Fig. 9 demonstrates the effectiveness of our approach in capturing and expressing the whole-body motion characteristics while ensuring the robot's feet remain fixed on the ground.

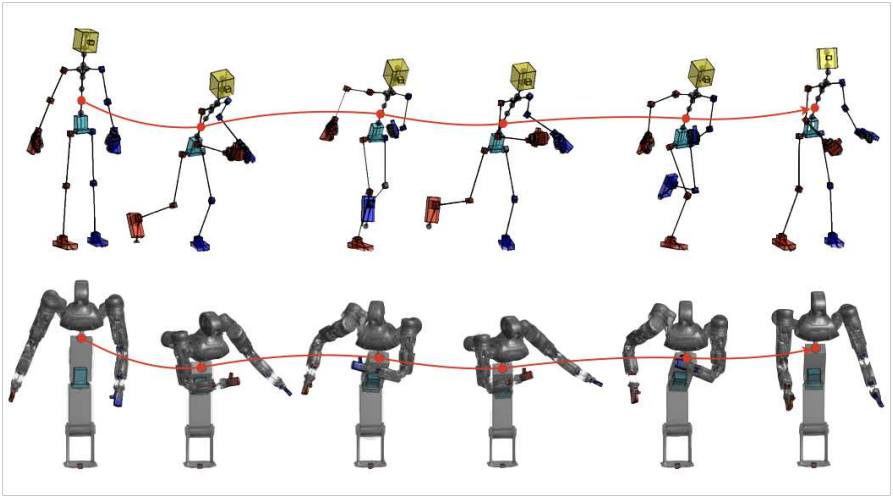

Fig. 9 Retargeting of the "Happy" motion from the Emotion MoCap to a robot using our proposed method. Despite fixing the robot's feet to the ground, the subtle shaking of the torso joint and the movements of the knee joints effectively capture the characteristics of the original motion, demonstrating the importance of considering both upper and lower body joint trajectories in the retargeting process.

The equality constraint $h(\cdot)$ in (Eq. 2b) ensures ground contact preservation by adjusting the target position of the foot joints, ensuring that both feet of the common-rig remain in contact with the ground throughout the motion. This constraint prevents the feet from penetrating the ground or floating in the air, thereby generating stable and physically plausible motions. The inequality constraint $g(\cdot)$ in (Eq. 2c) serves multiple purposes, including maintaining stability and balance, as well as preventing self-collisions. To maintain stability and balance, the positions of the knee and pelvis joints are adjusted so that both knees move in the same direction. Additionally, the common-rig's center of mass (COM) is constrained to stay within the contact polygon formed by the two feet. This can be formalized as:

$$\text{COM} \in \mathcal{P}$$

where $\mathcal{P}$ is the contact polygon defined by the robot's feet. The contact polygon can be represented by a set of linear inequalities:

$$a_i x + b_i y + c_i \geq 0, \quad i = 1, \ldots, n$$

where (a_i, b_i, c_i) are the parameters defining the i-th edge of the polygon, and n is the number of edges.

To prevent self-collisions, collision boundaries are established based on capsule information surrounding the mesh of the common-rig. The defined boundaries are as follows:

$$d + \epsilon < \mathrm{r}_1 + \mathrm{r}_2$$

The variables r_1 and r_2 represent the capsule radius corresponding to different parts of the common-rig, while d denotes the distance between the center lines of the capsules, and ϵ is the collision margin. If the sum of d and ϵ is less than the sum of r_1 and r_2, a collision flag is raised. To prevent self-collision, we iteratively solve the inverse kinematics problem, focusing on the direction that repels collision capsules from one another until no collisions are detected.

Finally, to ensure the common-rig always maintains a forward-facing orientation, an uprighting constraint is introduced to prevent the upper and lower bodies from moving separately, like in a situation where the upper body rotates one full lap. The uprighting constraint can be formalized as follows:

$$\hat{R}^{\text{rig}}_{\text{z},i=0}\mathbf{e}^{\text{rig}}_{\text{x},i=0} = \mathbf{e}_{\text{x}}$$

where $i = 0$ means index of the root, $\hat{R}^{\text{rig}}_{\text{z},i=0}$ denotes the z-axis orientation to align the x-axis of the root joint, $\mathbf{e}^{\text{rig}}_{\text{x},i=root}$, to the global x-axis, $\mathbf{e}_{\text{x}} = [1\ 0\ 0]^T$.

The post-rigging process, which incorporates these constraints, may result in an unsmooth motion trajectory, leading to abrupt speed changes and motion jitter. To address this, we repeatedly perform a motion trajectory smoothing task during the post-rigging process by applying the Gaussian Random Path method [49]. By considering the movements of the knee and pelvis joints, our approach effectively captures and expresses the whole-body motion characteristics while ensuring the robot's feet remain fixed to the ground. This allows us to generate expressive and feasible whole-body motions for the robot, even in the absence of complex controllers that consider dynamics.

4.3 Robot motion retargeting

In this section, we describe how the motion data, retargeted to the common-rig, is transferred to diverse robots. The robot motion retargeting process involves defining JOI for each robot, considering the differences in kinematic structure between the common-rig and the target robots. The JOI is set for corresponding joints on the robot, but they do not need to match the common-rig joints perfectly. For example, the JOI can be defined only for the upper body of robots like AMBIDEX, which have a unique lower body structure (Fig. 3C). The joint angles of the non-JOI joints are automatically calculated based on the IK solution for the JOI target positions and orientations.

Due to the varying link lengths among robots, the common-rig's joint information cannot be directly used as the IK target. Instead, we extract the direction vectors between connected joints from the common-rig and scale them using each robot's link lengths to determine the target positions for the robot's JOI. Furthermore, an adjustment process is performed on the robot's motion trajectory to ensure that it closely follows the original motion's trajectory. During motion retargeting, we also conduct self-collision checks and trajectory smoothing to ensure that the resulting robot motions are feasible.

4.3.1 Direction vector-based robot target pose

To retarget the common-rig's motion to diverse robots with different kinematic structures, we first define the JOI for each robot. The JOI is set for corresponding joints on the robot's body, such as the torso, neck, shoulders, elbows, and wrists for the upper body, and the pelvis, knees, ankles, and feet for the lower body. However, the JOI does not need to match the common-rig's joints exactly, allowing flexibility in handling robots with unique structures. By setting the JOI for the robot and the common rig, the common rig's motions can be automatically retargeted into motions feasible by the robot. We transfer joint information from the common-rig to the robot's joints selected as JOI. Specifically, we set the target positions and orientations in Cartesian space for the JOI and obtain the joint angle values for all the robot's joints by solving IK.

To consider the differences in link lengths between the common-rig and the robot, we extract the direction vectors between connected joints from the common-rig and scale them using the robot's link lengths. The target position of the robot joint j, connected to its parent joint i, is calculated as follows:

$$\mathrm{p}_j^{\mathrm{robot}} = \mathrm{p}_i^{\mathrm{robot}} + \mathrm{l}^{\mathrm{robot}} \cdot \mathbf{v}^{\mathrm{rig}}$$

where $\mathrm{l}^{\mathrm{robot}}$ is the link length of the robot, and $\mathbf{v}^{\mathrm{rig}}$ is the direction vector between the corresponding connected joints of the common-rig. $\mathrm{p}^{\mathrm{robot}}$ represents the robot joint positions, with $\mathrm{p}_0^{\mathrm{robot}}$ being the root position of the robot. We obtain a suitable target joint position for the robot's task space by scaling the direction vector using the robot's link length.

The entire robot target pose calculation process in our motion retargeting pipeline can be defined as follows:

$$\mathrm{T}_{1:L}^{\mathrm{robot}} = \mathrm{RT}\left(\xi^{\mathrm{robot}}, \xi^{\mathrm{rig}}, q_{1:L}^{\mathrm{rig}}\right)$$

where $\mathrm{T}_{1:L}^{\mathrm{robot}}$ is the target pose of the robot, derived by the function RT when given the robot's kinematic structure ξ^{robot}, the common-rig's kinematic structure ξ^{rig}, and the common-rig's joint angles $q_{1:L}^{\mathrm{rig}}$. The function RT computes the robot's target pose by combining the robot's link length information with the directional information obtained from the common-rig.

4.3.2 Robot motion trajectory adjustment

When performing a social gesture like clapping hands or making a heart symbol, the movement of the end-effector is crucial, and reaching a specific target position for the hand joints is essential. However, as the last joint in the kinematic tree, the end-effector's behavior is influenced not only by differences in kinematic structure between robots and humans (such as degrees of freedom and link length) but also by earlier parent joints. Consequently, simply reaching the target position in the task space does not guarantee the robot's proper execution of the social gesture.

To address this, we propose a process for adjusting the robot's motion trajectory, which retargets the source motion's end-effector trajectory in Cartesian space to the robot while preserving its shape. The original end-effector trajectory is scaled and biased using affine transformation parameters, serving as the final target for the robot's end-effector. To determine these parameters, we solve an optimization problem that minimizes the Euclidean distance between the source motion's end-effector trajectory and the robot motion's end-effector trajectory in Cartesian space. Our optimization problem is defined as follows:

$$\arg\min_{\theta} \frac{\sum(\left| f_\theta(\mathrm{T}_{1:L}^{\mathrm{mocap}}) - \mathrm{T}_{1:L}^{\mathrm{robot}} \right|_2)}{L}$$

$$\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}} = f_\theta(\mathrm{T}_{1:L}^{\mathrm{mocap}})$$

where f_θ represents the affine transformation function with parameters θ, and L is the length of the motion sequence. The optimization problem aims to find the optimal parameters θ that minimize the average Euclidean distance between the transformed source and robot motion trajectory. The robot's target position, $\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}}$, is obtained by applying the affine transformation to the source motion trajectory $\mathrm{T}_{1:L}^{\mathrm{mocap}}$.

Finally, the fine-tuning process is defined as:

$$\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}} = \mathrm{FT}(\mathrm{T}_{1:L}^{\mathrm{robot}}, \mathrm{T}_{1:L}^{\mathrm{mocap}})$$

where FT represents the fine-tuning function that takes the robot motion trajectory $\mathrm{T}_{1:L}^{\mathrm{robot}}$ and the source motion trajectory $\mathrm{T}_{1:L}^{\mathrm{mocap}}$ as inputs and returns the adjusted target trajectory $\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}}$ for the robot. To derive the feasible robot motion, $q_{1:L}^{\mathrm{robot}}$, we solve the following optimization problem:

$$\begin{aligned} \min_{q_{1:L}^{\mathrm{robot}}} \quad & d\left(\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}}, \mathrm{FK}\left(\xi^{\mathrm{robot}}, q_{1:L}^{\mathrm{robot}}\right)\right) \\ \text{s.t.} \quad & g(\xi^{\mathrm{robot}}, q_{1:L}^{\mathrm{robot}}) \geq 0 \end{aligned}$$

where $d(\cdot)$ represents the distance function, $\mathrm{FK}(\cdot)$ is the forward kinematics function, and $g(\cdot)$ represents the inequality constraints. The optimization problem aims to find the robot joint angles $q_{1:L}^{\mathrm{robot}}$ that minimize the distance between the target pose $\hat{\mathrm{T}}_{1:L}^{\mathrm{robot}}$ and the pose obtained by applying forward kinematics to the robot's joint angles while satisfying the inequality constraints. We employ numerical inverse kinematics to solve this optimization problem and obtain the final joint values for the robot. The obtained joint values ensure that the robot motion is feasible, avoids self-collisions, and considers gimbal lock.

4.3.3 Post-processing

In our approach, we utilize a direction vector-based methodology to adapt the common-rig motion to the robot's task space. However, the skeleton of a robot cannot be perfectly equivalent to the common-rig. For example, the neck joint of AMBIDEX is positioned in front of the root joint, unlike the common-rig. We employ a quaternion-based methodology to adjust the direction vector and establish the robot target pose

to handle these structural differences. The quaternion rotation offset Q represents the difference in direction vectors from the parent joint to the child joint when the robot and the common-rig are in their respective zero poses. This offset is determined by analyzing the kinematic differences between the robot and the common-rig skeletons.

The quaternion-based target pose adjustment is defined as:

$$\hat{\mathbf{v}}^{\text{rig}} = R_{\text{Q}}(\mathbf{v}^{\text{rig}})$$

where R_{Q} is a quaternion rotation operator that rotates the direction vector $\mathbf{v}^{\text{rig}}$ by the quaternion offset Q, resulting in the adjusted target direction vector $\hat{\mathbf{v}}^{\text{rig}}$.

The target position for the robot joint j, connected to its parent joint i, is then calculated using the adjusted target direction vector and the robot's link length:

$$\hat{\text{p}}_j^{\text{robot}} = \hat{\text{p}}_i^{\text{robot}} + \text{l}^{\text{robot}} \cdot \hat{\mathbf{v}}^{\text{rig}}$$

where $\hat{\text{p}}_j^{\text{robot}}$ and $\hat{\text{p}}_i^{\text{robot}}$ represent the target positions of the robot joints j and i, respectively, and l^{robot} is the link length of the robot.

To calculate the quaternion rotation offset Q, we first determine the direction vectors from the parent joint to the child joint in both the robot and the common-rig skeletons when they are in their respective zero poses. We then compute the quaternion that rotates the common-rig's direction vector to align with the robot's direction vector. This quaternion serves as the rotation offset Q. By applying the quaternion-based target pose adjustment, we obtain the final target positions for the robot joints, considering the structural differences between the robot and the common-rig skeletons. This post-processing step ensures that the retargeted motion is tailored to the specific robot structure, resulting in more accurate and realistic robot motions.

5 Data Availability

The datasets generated and analyzed during the current study are available from publicly accessible repositories. The motion capture data used in this study were obtained from the CMU MoCap [37] and Emotion MoCap [38] repositories[2], both of which are publicly accessible. Other motion data were generated from publicly available YouTube videos using FrankMoCap [34] and MHFormer [36].

6 Code Availability

The code for the proposed pipeline is available on GitHub at the following repository: https://github.com/sjchoi86/rmr-v2. This repository includes all necessary scripts and instructions to reproduce the results presented in this paper. The code is shared under the MIT License, with the restriction that any publications resulting from the use of this code must cite the original paper.

[2]The CMU Graphics Lab Motion Capture Database is available at http://mocap.cs.cmu.edu/. This database is free for use in research and education. The emotion motion capture database is available at https://physionet.org/content/kinematic-actors-emotions/2.1.0/ under the PhysioNet Restricted Health Data License 1.5.0. Access to this dataset requires user registration and signing of the specified Data Use Agreement.

7 Acknowledgements

This work was supported by the NAVER LABS and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University), No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI, and No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents). The authors express their gratitude to Seungjoon Yi and Joonyoung Kim from the Department of Electrical Engineering, Pusan National University, for their contributions to the JF2 motion experiments.

8 Author contributions

S.C. and T.J. proposed and developed the robust motion retargeting pipeline, which includes a common-rigging process to standardize humanoid robot models and a robot motion retargeting method to adapt human motions to robots. S.C. supervised the project and provided the overall research direction. T.J., T.B., and J.K. designed the experiments on real robots (AMBIDEX and THORMANG) and generated the necessary robot motion data. S.C. and T.J. analyzed the data and wrote the paper. K.C., J.O., and S.L. tested the retargeted motions on the AMBIDEX robot, including robot hardware control. O.D. and J.K. tested the retargeted motions on the THORMANG robot, also including robot hardware control. J.K. gave constructive advice for this work. All authors reviewed the manuscript.

9 Competing interests

The authors declare no competing interests.

References

[1] Tong, Y., Liu, H., Zhang, Z.: Advancements in humanoid robots: A comprehensive review and future prospects. IEEE/CAA Journal of Automatica Sinica **11**(2), 301–328 (2024)

[2] Cao, L.: Ai robots and humanoid ai: Review, perspectives and directions. arXiv preprint arXiv:2405.15775 (2024)

[3] LaViers, A.: Make robot motions natural. Nature **565**(7740), 422–424 (2019)

[4] Abe, N.: Beyond anthropomorphising robot motion and towards robot-specific motion: consideration of the potential of artist—dancers in research on robotic motion. Artificial Life and Robotics **27**(4), 777–785 (2022)

[5] Riley, M., Ude, A., Atkeson, C.G.: Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching. In: Proc. 2000 Workshop on Interactive Robotics and Entertainment, pp. 35–42 (2000). Citeseer

[6] Pollard, N.S., Hodgins, J.K., Riley, M.J., Atkeson, C.G.: Adapting human motion for the control of a humanoid robot. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2, pp. 1390–1397 (2002). IEEE

[7] Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., Ikeuchi, K.: Generating whole body motions for a biped humanoid robot from captured human dances. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 3, pp. 3905–3910 (2003). IEEE

[8] Kaplish, A., Yamane, K.: Motion retargeting and control for teleoperated physical human-robot interaction. In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 723–730 (2019). IEEE

[9] Schrum, M., Park, C.H., Howard, A.: Humanoid therapy robot for encouraging exercise in dementia patients. In: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 564–565 (2019). IEEE

[10] Gleicher, M.: Retargetting motion to new characters. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 33–42 (1998)

[11] Safonova, A., Pollard, N., Hodgins, J.K.: Optimizing human motion for the control of a humanoid robot. Proc. Applied Mathematics and Applications of Mathematics **78**, 18–55 (2003)

[12] Yamane, K., Hodgins, J.: Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2510–2517 (2009). IEEE

[13] Yamane, K., Anderson, S.O., Hodgins, J.K.: Controlling humanoid robots with human motion data: Experimental validation. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots, pp. 504–510 (2010). IEEE

[14] Ayusawa, K., Yoshida, E.: Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization. IEEE Transactions on Robotics **33**(6), 1343–1357 (2017)

[15] Dariush, B., Gienger, M., Arumbakkam, A., Goerick, C., Zhu, Y., Fujimura, K.: Online and markerless motion retargeting with kinematic constraints. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 191–198 (2008). IEEE

[16] Bin Hammam, G., Wensing, P.M., Dariush, B., Orin, D.E.: Kinodynamically consistent motion retargeting for humanoids. International Journal of Humanoid Robotics **12**(04), 1550017 (2015)

[17] Darvish, K., Tirupachuri, Y., Romualdi, G., Rapetti, L., Ferigo, D., Chavez, F.J.A., Pucci, D.: Whole-body geometric retargeting for humanoid robots. In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 679–686 (2019). IEEE

[18] Dariush, B., Gienger, M., Arumbakkam, A., Zhu, Y., Jian, B., Fujimura, K., Goerick, C.: Online transfer of human motion to humanoids. International Journal of Humanoid Robotics **6**(02), 265–289 (2009)

[19] Penco, L., Clément, B., Modugno, V., Hoffman, E.M., Nava, G., Pucci, D., Tsagarakis, N.G., Mouret, J.-B., Ivaldi, S.: Robust real-time whole-body motion retargeting from human to humanoid. In: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), pp. 425–432 (2018). IEEE

[20] Khalil, H., Coronado, E., Venture, G.: Human motion retargeting to pepper humanoid robot from uncalibrated videos using human pose estimation. In: 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), pp. 1145–1152 (2021). IEEE

[21] Shon, A., Grochow, K., Hertzmann, A., Rao, R.P.: Learning shared latent structure for image synthesis and robotic imitation. Advances in neural information processing systems **18** (2005)

[22] Levine, S., Wang, J.M., Haraux, A., Popović, Z., Koltun, V.: Continuous character control with low-dimensional embeddings. ACM Transactions on Graphics (TOG) **31**(4), 1–10 (2012)

[23] Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., Chen, B.: Skeleton-aware networks for deep motion retargeting. ACM Transactions on Graphics (TOG) **39**(4), 62–1 (2020)

[24] Reda, D., Won, J., Ye, Y., Panne, M., Winkler, A.: Physics-based motion retargeting from sparse inputs. Proceedings of the ACM on Computer Graphics and Interactive Techniques **6**(3), 1–19 (2023)

[25] Yamane, K., Ariki, Y., Hodgins, J.: Animating non-humanoid characters with human motion data. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 169–178 (2010)

[26] Lawrence, N.: Gaussian process latent variable models for visualisation of high dimensional data. Advances in neural information processing systems **16** (2003)

[27] Kim, T., Lee, J.-H.: C-3po: Cyclic-three-phase optimization for human-robot

motion retargeting based on reinforcement learning. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 8425–8432 (2020). IEEE

[28] Choi, S., Pan, M., Kim, J.: Nonparametric motion retargeting for humanoid robots on shared latent space. Proceedings of Robotics: Science and Systems (R: SS) (2020)

[29] Choi, S., Song, M.J., Ahn, H., Kim, J.: Self-supervised motion retargeting with safety guarantee. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 8097–8103 (2021). IEEE

[30] Vicon Systems. https://www.vicon.com/

[31] OptiTrack-Motion Capture. https://optitrack.com/

[32] Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)

[33] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Yong, M., Lee, J., et al.: Mediapipe: a framework for building perception pipelines (2019). arXiv preprint arXiv:1906.08172 (1906)

[34] Rong, Y., Shiratori, T., Joo, H.: Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1749–1759 (2021)

[35] Fang, H.-S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.-L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[36] Li, W., Liu, H., Tang, H., Wang, P., Van Gool, L.: Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13147–13156 (2022)

[37] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/

[38] Zhang, M., Yu, L., Zhang, K., Du, B., Zhan, B., Chen, S., Jiang, X., Guo, S., Zhao, J., Wang, Y., *et al.*: Kinematic dataset of actors expressing emotions. Scientific data **7**(1), 292 (2020)

[39] Goodall, C.: Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society: Series B (Methodological) **53**(2), 285–321 (1991)

[40] Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern

Recognition, pp. 3383–3393 (2021)

[41] Liu, J., Shi, M., Chen, Q., Fu, H., Tai, C.-L.: Normalized human pose features for human action video alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11521–11531 (2021)

[42] Gärtner, E., Andriluka, M., Coumans, E., Sminchisescu, C.: Differentiable dynamics for articulated 3d human motion reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13190–13200 (2022)

[43] Gärtner, E., Andriluka, M., Xu, H., Sminchisescu, C.: Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13106–13115 (2022)

[44] Yang, Z., Cai, Z., Mei, H., Liu, S., Chen, Z., Xiao, W., Wei, Y., Qing, Z., Wei, C., Dai, B., *et al.*: Synbody: Synthetic dataset with layered human models for 3d human perception and modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 20282–20292 (2023)

[45] Henze, B., Ott, C., Roa, M.A.: Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3253–3258 (2014). IEEE

[46] Gomes, W., Radhakrishnan, V., Penco, L., Modugno, V., Mouret, J.-B., Ivaldi, S.: Humanoid whole-body movement optimization from retargeted human motions. In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 178–185 (2019). IEEE

[47] Penco, L., Hoffman, E.M., Modugno, V., Gomes, W., Mouret, J.-B., Ivaldi, S.: Learning robust task priorities and gains for control of redundant robots. IEEE Robotics and Automation Letters **5**(2), 2626–2633 (2020)

[48] Kajita, S., Hirukawa, H., Harada, K., Yokoi, K.: Introduction to Humanoid Robotics vol. 101. Springer, Heidelberg (2014)

[49] Choi, S., Lee, K., Oh, S.: Gaussian random paths for real-time motion planning. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1456–1461 (2016). IEEE

[50] Choi, S., Kim, J.: Towards a natural motion generator: A pipeline to control a humanoid based on motion data. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4373–4380 (2019). IEEE

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- movie1motionretargetingresults.mp4
- movie2motionretargetingpipeline.mp4