A Low-cost & Real-time Motion Capture System

Anargyros Chatzitofis[1], Georgios Albanis[2], Nikolaos Zioulis[3], Spyridon Thermos[1]

[1]Codewheel {chatzitofis, spthermo}@codewheel.eu
[2]Dept. of Informatics and Telecommunications, University of Thessaly galmpanis@uth.gr
[3]Independent Researcher nzioulis@gmail.com

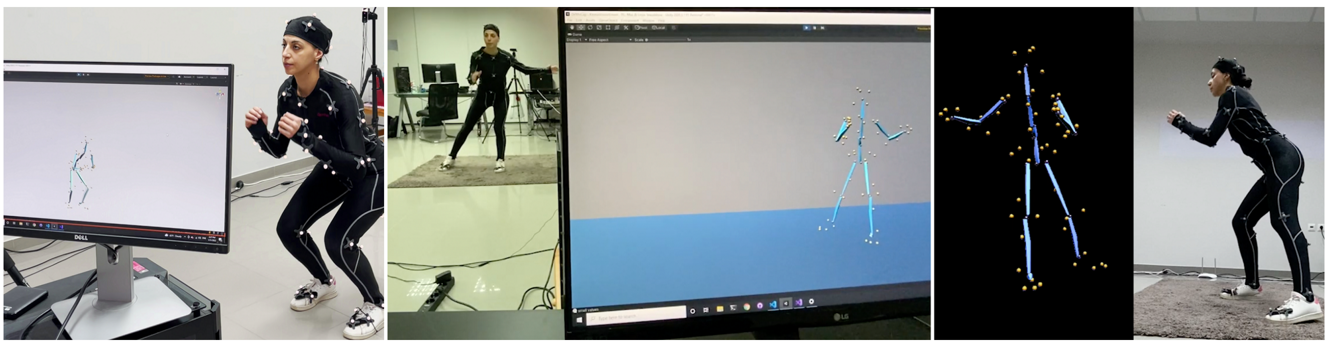

Figure 1. We propose a system for marker-based MoCap using a sparse set (3 in our captures) of commodity RGB-D sensors. Our system is an adapted version of the DeMoCap model [5], operating in real-time and offering MoCap pre-visualization, as well as real-time denoising due to its data-driven backend. Compared to traditional marker-based systems, its denoising nature offers robustness to marker placement, and compared to markerless systems, it offers metric-scale (compared to monocular approaches), and more consistent (compared to multiview) results.

Abstract

Traditional marker-based motion capture requires excessive and specialized equipment, hindering accessibility and wider adoption. In this work, we demonstrate such a system but rely on a very sparse set of low-cost consumer-grade sensors. Our system exploits a data-driven backend to infer the captured subject's joint positions from noisy marker estimates in real-time. In addition to reduced costs and portability, its inherent denoising nature allows for quicker captures by alleviating the need for precise marker placement and post-processing, making it suitable for interactive virtual reality applications.

1. Introduction & Prior Art

Motion Capture (MoCap) is realized as the human-centric technology[1] that aims to digitize human motion, and thus, performances, and is primarily used for analysis (*i.e.* clinical, athletic or artistic) [7, 12, 15, 24] and content creation (*i.e.* games, films, simulation) [2, 13, 21, 26]. It comes in many variants, depending on the equipment and technology stack used, spanning marker-based and markerless optical, or wearable sensor-based, with each one carrying its own set of advantages and disadvantages [14, 18]. While currently the marker-based optical MoCap systems are considered as the most accurate solution, the flexibility and cost reduction of markerless or inertial alternatives has increased their use in domains where high accuracy is not strictly necessary. Additionally, the emergence of virtual/mixed/augmented reality (VR/MR/AR) applications is expected to further boost the need for affordable, easy-to-use, portable and flexible MoCap systems.

Interestingly, the technological evolution brought forth by data-driven technologies has been more impacting on the markerless [8] and inertial [10] methods than on marker-based optical ones, where it has mostly been used to repair [16], denoise [25] and clean [3] their captures. Only a few, recent works approach motion capture data solving [6, 9], not evaluated on data captured with low-end, noisy capturing systems though. Still, the use of markers

[1]Even though MoCap technology can be used to capture other subjects like animals or machines (drones, robotic arms, etc.), our focus in this manuscript lies solely on human capture.

comes with a set of advantages and possibilities not available in markerless or inertial systems, such as the addition of props, the physical grounding of the captures, the precise and robust calculation of the joint rotations with the use of marker positions, and the adaptation to different context or increased accuracy (*e.g.* higher quality foot captures). Even though hybrid systems have been recently introduced [1], no progress has been reported on lower-cost and/or data-driven marker-based systems.

Addressing this precise gap, this work presents a working and demonstratable system for real-time marker-based MoCap using a sparse set of commercial-grade sensors. Instrumental to such system that offers an order of magnitude equipment cost reduction, is the use of data-driven technology to develop a neural marker human motion prior model [5]. This allows our system to infer joint positions from noisy and low quality marker estimates in a single-shot, effectively performing clean-up and denoising simultaneously, correcting marker artifacts like ghosting and missing information. The summary of contributions that drive the demonstration of this system are the following:

- The design and the development of a multi-sensor multi-view system integrating the aforementioned data-driven model that is easy-to-use and quick-to-deploy.
- The adaptation of the data-driven model to a new sensor and the model's real-time and low-latency performance (inference).

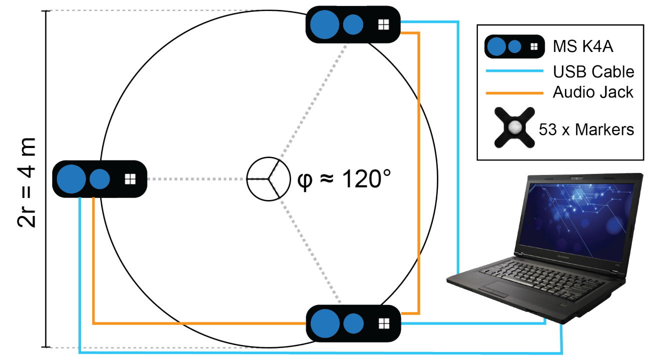

Figure 2. The proposed MoCap system comprises hardware (HW) and software (SW) components. From a HW perspective, a minimal set of tripod-mounted commodity sensors are required (3 Microsoft K4A shown), connected with a workstation that handles the processing (cyan links). Typical outwards-in placement requires them to be equidistantly placed from an angular perspective around a pre-determined radius ($r = 2m$ in this case). Additionally, HW synchronization cables inter-connect the sensors (orange links). Finally, 53 retro-reflective markers are also required to be placed onto the subject to be captured.

2. System

In this section we present details about our system's design and functionality, spanning both hardware and software. Our core motion capture technology is derived from a recent work for data-driven marker-based motion capture [5]. The first steps towards an operational real-time MoCap system are: the development of efficient data acquisition components (discussed in Sec. 2.1 along with the main goals and overall system design); the robust estimation of the marker positions (described in Sec. 2.2); and the spatio-temporal alignment of such multiple sensors, described in Sec. 2.3. A critical step follows, namely the adaptation of a staged markers-to-joints model [5] (Sec. 2.4), comprising multiple CNN stacks to low-latency and real-time rate performances. Finally, considering that the training data are fixed, and the system is built around a new sensor type, the model training regime also needs to be adapted to overcome any domain biases (Sec. 2.5).

2.1. System Design

The design of the presented marker-based MoCap system is dictated by a set of requirements: **i)** the detection of retro-reflective markers (sensing), **ii)** the operation in real-time rates (processing latency minimization), and **iii)** the usability (easy-to-deploy/setup/use effectively). For the latter, we opt to use commodity sensing hardware and minimize the amount of deployed sensors. To enable image-based retro-reflective marker detection, the selected sensors should be capable of projecting infrared (IR) light into the scene, and then capturing it back. The efficient minimization of sensors (and thus, viewpoints), will be driven by the acquisition of 3D information straight from the sensors, which is another important design choice. Additionally, given the nature of motion capture, it is necessary to be able to precisely synchronize the deployed sensors' acquisition, without requiring excessive hardware. Taking all these into account, and given the discontinuation of the Intel RealSense (RS2) series, and the pending availability of active IR capable OAK-D sensors, we present our system using the Microsoft Kinect Azure (K4A) sensor. In particular, we use a small, sparse set of these sensors deployed, spanning the range of $3-6$. A schematic representation of the actual capturing setting used for this demo is depicted in Fig. 2.

2.2. Marker Acquisition

Each sensor $s \in \{1, ..., S\}$ acquires time- and pixel-aligned infrared $\mathbf{I}_s(\mathbf{p}) \in \mathbb{R}$ and depth $\mathbf{D}_s(\mathbf{p}) \in \mathbb{R}$ frames, both carrying 16-bit information, with $\mathbf{p} := (u, v) \in \Omega$, and Ω being the image domain grid defined with an image resolution of W width and H height. Given that retro-reflective markers bounce light back directly to the source, and that K4A is a time-of-flight sensor that projects infrared light

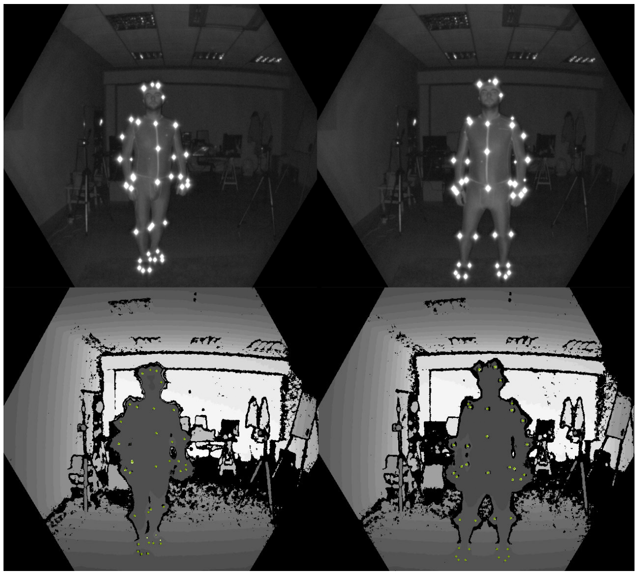

Figure 3. Two examples of marker detection results using the K4A sensor. On the top row, the IR images are depicted where the retro-reflective markers are clearly identified. On the bottom row, the IR detections (green stars) are overlayed on top of the depth maps, inside the empty/invalid depth measurement regions that correspond to the retro-reflective "blind" spots for the K4A sensor.

into the scene and captures the bounced back light from its infrared camera, retro-reflective markers appear as excessively bright in $\mathbf{I}_s$ as the corresponding areas' signal amplitude is maximal (see Fig. 3 top). Consequently, they are very easy to be detected via straightforward thresholding of the IR image, with a high-level of robustness with respect to the choice of the threshold.

Using connected components analysis on the thresholded images, we extract the N detected markers and represent them via their centroids $\mu_n^s \in \Omega, n \in \{1, ..., N\}$. While these are 2D detections, the availability of depth information allows us to lift them to 3D marker estimates. Even though the centroid corresponding depth information is missing as the maximal signal amplitude does not allow for estimating it, the spherical nature of the markers allows for some light to scatter into the scene at the areas where the marker's surface is close to perpendicular to the camera. This creates a depth "ring" around the marker, whose depth values approximate that of the marker's detected centroid. We extract the median of the one-ring neighbourhood of depth values around the detected marker blob, ensuring an unbiased and denoised estimate, which is then lifted to a 3D marker detection $\mathbf{m}_n^s \in \mathbb{R}^3$ using the camera parameters comprising the intrinsics matrix $\mathbf{K}_s$ and the distortion coefficients $\mathbf{d}_s \in \mathbb{R}^5$. The result of the marker detection is depicted in Fig. 3 (bottom).

2.3. Multi-sensor Spatio-temporal Alignment

The 3D marker estimates $\mathbf{m}_n^s$ acquired by each sensor s are defined on the sensor's local coordinate system and suffer from occlusions as each viewpoint partially observes the captured performance. To effectively fuse these marker estimates from all viewpoints, we need to ensure their spatial and temporal alignment. For the latter, we resort to the selected sensor's hardware synchronization and additionally apply a small offset on the order of microseconds[2]. This is to overcome the multi-path interference of multiple IR projectors illuminating the same scene simultaneously, but still benefit from the high-quality temporal synchronization of all sensors, a necessity for capturing moving subjects.

To recover the 6DOF pose $\mathbf{T^s} := \begin{bmatrix} \mathbf{R}_s & \mathbf{t}_s \\ \mathbf{0} & 1 \end{bmatrix}$ of all sensors in a common, global coordinate system, we employ a quick and effortless approach. Using a moving wand with a single marker attached on its tip, we extract multiple 2D and 3D correspondences of a single marker detection, μ_k^s and $\mathbf{m}_k^s$, respectively. Our approach is greedy and considers only cases where a single marker is detected across all viewpoints. Since we also extract the local 3D marker estimates $\mathbf{m}_n^s$ in addition to the projections μ_n^s, we first perform a pairwise alignment with respect to a chosen reference viewpoint $s_{ref} = 1$ using the unscaled Umeyama algorithm [17] and the 3D correspondences $\mathbf{m}_k^s$. This provides us with an adequate initial estimation for each sensor's pose $\mathbf{T}_{init}^s, s \in \{2, ..., S\}$. We then use the projection constraints μ_k^s to perform graph-based sparse bundle adjustment [11] for the sensor poses, except the one s_{ref} used as a reference, keeping it fixed as the identity pose, as well as fixing the 3D marker estimates originating from this reference viewpoint. This step quickly refines the pairwise esti-

[2] Specifically, $160\mu s$ as explained in the K4A documentation.

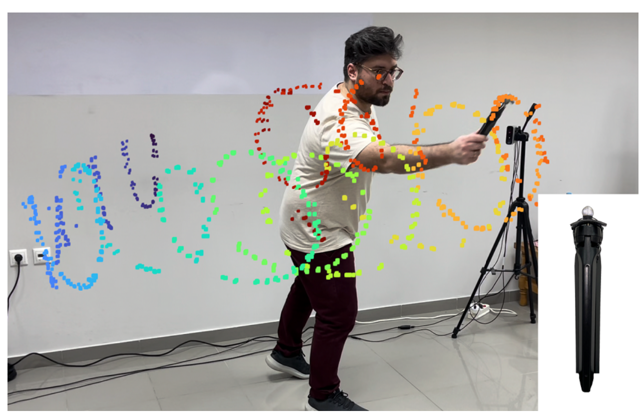

Figure 4. A visualization of the simple multi-sensor extrinsics calibration process. Using a single-marker wand (bottom right), the user only needs to freely move the wand creating a trajectory of correspondences (color-coded based on elapsed time).

mates, to the final globally optimized $\mathbf{T}^s$. Using these, we fuse all local marker estimates from each sensor s, resulting into a 3D marker cloud $\hat{\mathbf{m}} = \bigcup_{s=1,n=1}^{S,N} \mathbf{T}^s \mathbf{m}_n^s$, which is the input to our model. A sample aligned wand trajectory as captured from all viewpoints is illustrated in Fig. 4.

2.4. Real-time Inference

We adapt the staged DeMoCap model [5] to achieve real-time run-time rates and minimize processing latency. The original model uses a staged markers-to-joints approach, where $2\times$ HRNet [20] models are used in a cascade, with the first one predicting markers using a 4 branch/stage HRNet, and then encodes the predicted markers into joints using the second model, again via 4 branches/stages of high-resolution modules. While one approach to reduce the computational complexity would be to reduce the number of stacks, or retrain a single HRNet model to predict joints from noisy markers, these approaches would sacrifice representation power and accuracy for run-time at a sub-optimal trade-off. Instead, we leverage the recently presented Lite-HRNet [22] network that offers a more balanced trade-off between model performance and run-time. Since the staged markers-to-joints approach improves the quality of the results, we retrain the original model using Lite-HRNets instead of traditional HRNets it was presented with. The resulting "DeMoCap-Lite" model is capable of real-time inference at the sensor acquisition rate, with evidence presented in our supplementary video[3].

2.5. Sensor Adaptation

The original DeMoCap model [5] was trained with marker data obtained from the RS2 IR and depth streams and supervised with the Vicon MoCap marker and joint data [4]. Developing a system on top of the K4A sensors means that the model input data distribution will be shifted compared to that which the model was trained on. This data discrepancy manifests in two ways, first on the 2D marker detection level and, second, on the depth estimates used to lift the 2D marker detections to the fused 3D marker cloud input of the model. RS2 estimates depth through active stereo, projecting a dotted IR pattern into the scene which is sparser than the K4A IR projector. As seen in Fig. 5, this results into non-continuous high amplitude blobs that depict a single marker, which alters the post-thresholding output as well as the 2D marker centroid detection robustness. Additionally, RS2 depth is far more noisy than K4A depth, even when using stricter stereo matching thresholds and at closer distances, as also indicated in Fig. 5. Overall, the original model was trained on far noisier data than what is captured by the K4A sensors. Still, the K4A data are noisier than the high-quality Vicon data from a systematic error noise perspective, but they also suffer from occlusions and ghosting,

[3]https://www.codewheel.eu/cvpr2022/video

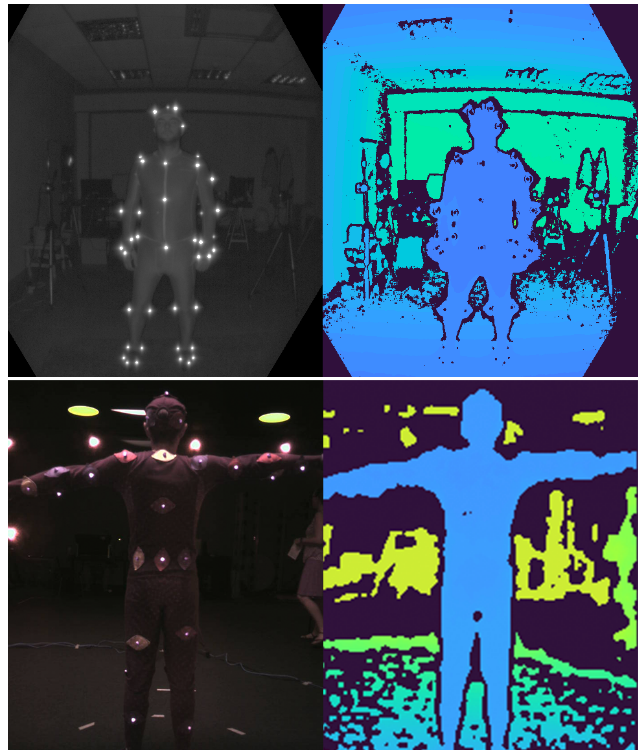

Figure 5. An illustration of the domain gap between the K4A sensor (top) and the RS2 sensor (bottom). On the left, the input IR images are depicted, with the K4A blobs being of higher-quality, while on the right, the input depth images are visualized, with the K4A depth map exhibiting the invalid values at the markers' positions. On the contrary, RS2 misses markers due to its sparser projected dot pattern, hindering their robust detection (*e.g.* thighs), and offers much more noisy depth, especially at the high amplitude marker regions that contain no informative features for stereo-matching. Further, the one-ring effect is also observed, where the measurements next to the invalid area preserve the surface's depth.

an information type of noise which exists in the RS2 data, but not in the Vicon ground-truth. To overcome these issues and increase the model's performance on the new sensor without capturing new data, we employ a curriculum learning training approach that seeks to control the level of noise that the model is trained on. During the initial phases of training (first 30 epochs), our DeMoCap variant is trained only with Vicon marker data as input, essentially learning to autoencode high-quality marker input, and infer the body structure from them. Then, for the next training phase (epochs $31 \rightarrow 80$), we add systematic and information noise to the Vicon data, steering the model towards learning to denoise the inputs. For the systematic noise we use Gaussian noise $\mathcal{N}(0, \boldsymbol{\sigma})$ on the marker 3D positions, with a randomly uniform sampled deviation $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$, where $\sigma = \mathcal{U}(0.5cm, 1.5cm)$. Additionally, for the information noise, we randomly remove up to 8 markers, and addition-

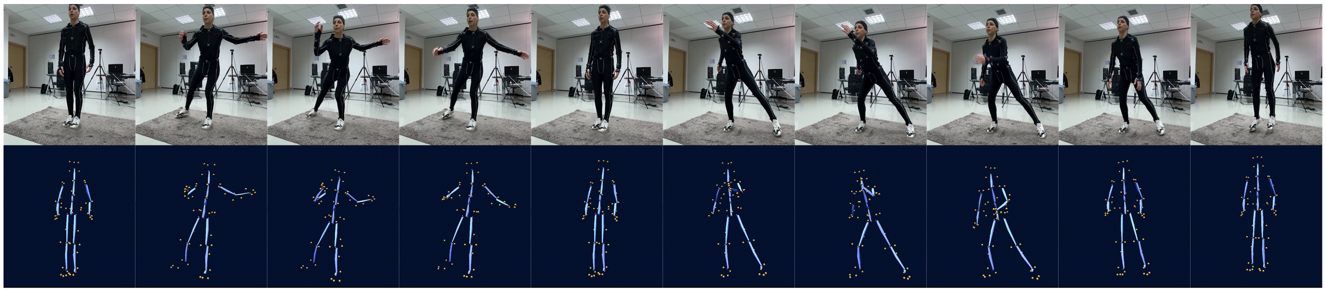

Figure 6. A representative example of a captured motion sequence: (top) the physical scene and the actress as perceived by an observer; (bottom) the real-time output of our system rendered as 3D skeleton along with markers frames.

ally randomly generate up to 8 new markers around existing markers, with their position's offset drawn from a Gaussian distribution $\mathcal{N} = (0, \mathcal{U}(1.5cm, 5.0cm))$. Essentially, these add synthetic ghosting and occlusion artifacts on the input. Finally, for the last training stage (last 40 epochs), we include the RS2 data to supplement the noisy Vicon data, creating a mix of input noise that prevents the model from focusing on a specific distribution, and aligning it better with the higher quality K4A input distribution.

The total average processing time per frame, spanning from the per-sensor capturing (3 viewpoints) up to the human motion rendering, is $30ms$ when running on a Laptop with an RTX 2080 Ti GPU and an Intel i7 CPU, resulting in low-latency motion capture at 30 FPS, equal to the sensor acquisition frame rate.

3. Results & Discussion

The presented system allows for the motion capture of single subject performances using affordable and lightweight equipment. It operates in real-time, enabling pre-visualization and live inspection of the performance results. Compared to monocular markerless approaches, our system can obtain more robust, metric-scale captures with a minimal equipment/costs overhead. Fig. 6 shows the inputs, accompanying with color information, and the captured motion of the actor's performance in time. More results are available in the supplementary video[3], with an extensive quantitative analysis for the original model available in [5]. Further, the system has been tested with various performances, proving its generalization to motions outside the training dataset, including dancing, sports, and casual/social ones. It should be noted that for all qualitative results, the system performs per-frame inference *without using* any temporal information or other constraints. Furthermore, another notable trait is its sensor agnostic nature, which is evident when considering that the training data have not been captured by K4A sensors. Finally, its data-driven backend offers high level of robustness with respect to the markers' placement on each subject, speeding up capturing workflows and reducing repeat captures due to sub-optimal marker placements.

Therefore, future work will focus on integrating temporal constraints/tracking, similar to [23], as well as a body structure calibration step to enforce the consistency of the estimated bone lengths in an explicit way. Additionally, we plan to scale up our system for covering larger capturing areas and/or supporting multiple subjects, yet the current system design is already able to facilitate these changes.

Nonetheless, there is a set of limitations which will require deeper modifications. Reliance on markers is one of the aforementioned limitation, which is partially the reason why markerless methods are recently surfacing. Still, exploiting available data, a possible direction would be to reduce the number of markers, similar to how inertial MoCap systems are starting to reduce the number of deployed IMUs [19]. Likewise, a fusion of color and infrared information may enable the development of hybrid, *i.e.* marker-based and markerless, systems targeting the reduction of the markers attached on the subjects, while preserving accuracy and estimating metric-scale outputs. Such systems are commercially available nowadays, but, to the best of our knowledge, none of them exploits extensively the advances of data-driven techniques.

4. Acknowledgements

A. Chatzitofis and S. Thermos would like to acknowledge financial support by the RIF through the KinesisVision project under contract PRE-SEED/0719/0119 and HUB-CAP H2020 EU project under contract 2021/1578786.

References

[1] Markerless software from Theia in Vicon Nexus: Motion capture software, 2021.

[2] Tanine Allison. More than a man in a monkey suit: Andy serkis, motion capture, and digital realism. *Quarterly Review of Film and Video*, 28(4):325–341, 2011.

[3] Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, and Ariel Shamir. Self-similarity analysis for motion capture cleaning. *Computer graphics forum*, 37(2):297–309, 2018.

[4] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zioulis, Shishir Subramanyam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas, et al. HUMAN4D: A human-centric multimodal dataset for motions and immersive media. *IEEE Access*, 8:176241–176262, 2020.

[5] Anargyros Chatzitofis, Dimitrios Zarpalas, Petros Daras, and Stefanos Kollias. DeMoCap: Low-cost marker-based motion capture. *International Journal of Computer Vision*, 129(12):3338–3366, 2021.

[6] Kang Chen, Yupan Wang, Song-Hai Zhang, Sen-Zhe Xu, Weidong Zhang, and Shi-Min Hu. MoCap-Solver: A neural solver for optical motion capture data. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[7] Henrique Galvan Debarba, Marcelo Elias de Oliveira, Alexandre Lädermann, Sylvain Chagué, and Caecilia Charbonnier. Augmented reality visualization of joint movements for physical examination and rehabilitation. In *Proc. IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 537–538, 2018.

[8] Yann Desmarais, Denis Mottet, Pierre Slangen, and Philippe Montesinos. A review of 3D human pose estimation algorithms for markerless motion capture. *Computer Vision and Image Understanding*, 212:103275, 2021.

[9] Daniel Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[10] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37(6):185:1–185:15, Nov. 2018.

[11] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[12] Marianna Liparoti, Emahnuel Troisi Lopez, and Valeria Agosti. Motion capture system: A useful tool to study cyclist's posture. *Journal of Physical Education and Sport*, 20(4):2364–2367, 2020.

[13] Alberto Menache. *Understanding motion capture for computer animation and video games*. Morgan kaufmann, 2000.

[14] Matteo Menolotto, Dimitrios-Sokratis Komaris, Salvatore Tedesco, Brendan O'Flynn, and Michael Walsh. Motion capture technology in industrial applications: A systematic review. *Sensors*, 20(19):5687, 2020.

[15] Siriporn Noiumkar and Suwich Tirakoat. Use of optical motion capture in sports science: A case study of golf swing. In *Proc. IEEE International Conference on Informatics and Creative Multimedia (ICICM)*, pages 310–313, 2013.

[16] Maksym Perepichka, Daniel Holden, Sudhir P Mudur, and Tiberiu Popa. Robust marker trajectory repair for MOCAP using kinematic reference. In *Motion, Interaction and Games*, pages 1–10. 2019.

[17] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[18] Eline van der Kruk and Marco M Reijne. Accuracy of human motion capture systems for sport applications; state-of-the-art review. *European journal of sport science*, 18(6):806–819, 2018.

[19] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3D human pose estimation from sparse IMUs. In *Computer Graphics Forum*, volume 36, pages 349–360, 2017.

[20] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[21] Katsu Yamane and Jessica Hodgins. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2510–2517, 2009.

[22] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-HRNet: A lightweight high-resolution network. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10440–10450, 2021.

[23] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4D association graph for real time multi-person motion capture using multiple video cameras. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1324–1333, 2020.

[24] Mei Zhu. Dance basic training methods based on motion capture technology. In *Proc. International Conference on Computers, Information Processing and Advanced Education*, pages 1278–1281, 2021.

[25] Yongqiong Zhu. Denoising method of motion capture data based on neural network. *Journal of Physics: Conference Series*, 1650(3):032068, 2020.

[26] Victor Brian Zordan and Jessica K Hodgins. Motion capture-driven simulations that hit and react. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 89–96, 2002.