XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera

DUSHYANT MEHTA, MPI For Informatics, Saarland Informatics Campus
OLEKSANDR SOTNYCHENKO, MPI For Informatics, Saarland Informatics Campus
FRANZISKA MUELLER, MPI For Informatics, Saarland Informatics Campus
WEIPENG XU, MPI For Informatics, Saarland Informatics Campus
MOHAMED ELGHARIB, MPI For Informatics, Saarland Informatics Campus
PASCAL FUA, École Polytechnique Fédérale de Lausanne
HANS-PETER SEIDEL, MPI For Informatics, Saarland Informatics Campus
HELGE RHODIN, University of British Columbia, École Polytechnique Fédérale de Lausanne
GERARD PONS-MOLL, MPI For Informatics, Saarland Informatics Campus
CHRISTIAN THEOBALT, MPI For Informatics, Saarland Informatics Campus

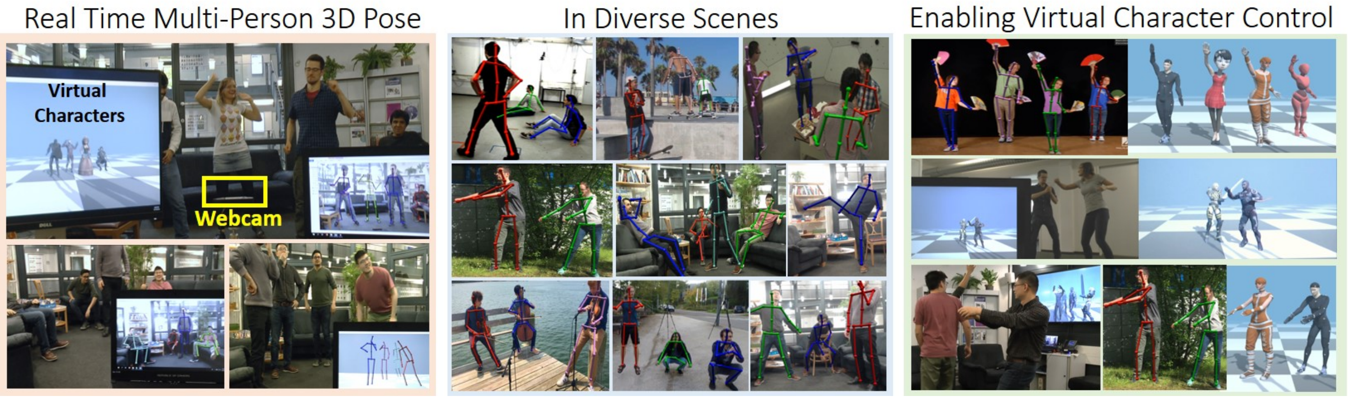
Real Time Multi-Person 3D Pose
In Diverse Scenes
Enabling Virtual Character Control
Virtual Characters
Webcam

Fig. 1. Our real-time monocular RGB based 3D motion capture provides temporally coherent estimates of the full 3D pose of multiple people in the scene, handling occlusions and interactions in general scene settings, and localizing subjects relative to the camera. Our design allows the system to handle large groups of people in the scene with the run-time only minimally affected by the number of people in the scene. Our method yields full skeletal pose in terms of joint angles, which can readily be employed for live character animation. Some images courtesy [KNG Music 2019], [Music Express Magazine 2013a]. 3D characters from Mixamo [Adobe 2020].

We present a real-time approach for multi-person 3D motion capture at over 30 fps using a single RGB camera. It operates successfully in generic scenes which may contain occlusions by objects and by other people. Our method operates in subsequent stages. The first stage is a convolutional neural network (CNN) that estimates 2D and 3D pose features along with identity assignments for all visible joints of all individuals. We contribute a new architecture for this CNN, called *SelecSLS Net*, that uses novel selective long and short range skip connections to improve the information flow allowing for a drastically faster network without compromising accuracy. In the second stage, a fully-connected neural network turns the possibly partial (on account of occlusion) 2D pose and 3D pose features for each subject into a complete 3D pose estimate per individual. The third stage applies space-time skeletal model fitting to the predicted 2D and 3D pose per subject to further reconcile the 2D and 3D pose, and enforce temporal coherence. Our method returns the full skeletal pose in joint angles for each subject. This is a further key distinction from previous work that do not produce joint angle results of a coherent skeleton in real time for multi-person scenes. The proposed system runs on consumer hardware at a previously unseen speed of more than 30 fps given 512x320 images as input while achieving state-of-the-art accuracy, which we will demonstrate on a range of challenging real-world scenes.

CCS Concepts: • **Computing methodologies → Motion capture**; *Computer vision*; *Neural networks*.

Additional Key Words and Phrases: human body pose, motion capture, real-time, monocular, RGB

Authors' addresses: Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Hans-Peter Seidel, Gerard Pons-Moll, and Christian Theobalt, MPI For Informatics, Saarland Informatics Campus; Pascal Fua, École Polytechnique Fédérale de Lausanne; Helge Rhodin, University of British Columbia.

0730-0301/2020/7-ART82 $15.00
https://doi.org/10.1145/3386569.3392410

ACM Reference Format:
Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera. *ACM Trans. Graph.* 39, 4, Article 82 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392410

1 INTRODUCTION

Optical human motion capture is a key enabling technology in visual computing and related fields [Chai and Hodgins 2005; Menache 2010; Starck and Hilton 2007]. For instance, it is widely used to animate virtual avatars and humans in VFX. It is a key component of many man-machine interfaces and is central to biomedical motion analysis. In recent years, computer graphics and computer vision researchers have developed new motion capture algorithms that operate on ever simpler hardware and under far less restrictive constraints than before. These algorithms do not require special body suits, dense camera arrays, in-studio recording, or markers. Instead, they only need a few calibrated cameras to capture people wearing everyday clothes outdoors, e.g. Elhayek et al. [2016]; Fang et al. [2018]; Huang et al. [2017a]; Kanazawa et al. [2018]; Mehta et al. [2017b]; Omran et al. [2018]; Pavlakos et al. [2019]; Rhodin et al. [2016b]; Stoll et al. [2011]; Xiang et al. [2019]. The latest approaches leverage the power of deep neural networks to capture 3D human pose from a single color image, opening the door to many exciting applications in virtual and augmented reality. Unfortunately, the problem remains extremely challenging due to depth ambiguities, occlusions, and the large variety of appearances and scenes.

More importantly, most methods fail under occlusions and focus on a single person. Some recent methods instead focus on the egocentric setting [Rhodin et al. 2016a; Tome et al. 2019; Xu et al. 2019]. Single person tracking in the outside-in setting (non-egocentric) is already hard and starkly under-constrained; multi-person tracking is incomparably harder due to mutliple occlusions, challenging body part to person assignment, and is computationally more demanding. This presents a practical barrier for many applications such as gaming and social VR/AR, which require tracking *multiple people* from low cost sensors, and in *real time.*

Prior work on multi-person pose estimation runs at best at interactive frame rates (10-15 fps) [Dabral et al. 2019; Rogez et al. 2019] or offline [Moon et al. 2019], and produces per-frame joint position estimates which cannot be directly employed in many end applications requiring joint angle based avatar animations.

We introduce a real-time algorithm for motion capture of multiple people in common interaction scenarios using a single color camera. Our full system produces the skeletal joint angles of multiple people in the scene, along with estimates of 3D localization of the subjects in the scene relative to the camera. Our method operates at more than 30 frames-per-second and delivers state-of-the-art accuracy and temporal stability. Our results are of a similar quality as commercial depth sensing based mocap systems.

To this end, we propose a new pose formulation and a novel neural network architecture, which jointly enable real-time performance, while handling inter-person and person-object occlusions. A subsequent model-based pose fitting stage produces temporally stable 3D skeletal motions. Our pose formulation uses two deep neural network stages that perform local (per body joint) and global (all body joints) reasoning, respectively. *Stage I* is fully convolutional and jointly reasons about the 2D and 3D pose for all the subjects in the scene at once, which ensures that the computational cost does not increase with the number of individuals. Since *Stage I* handles the already complex task of parsing the image for body parts, as well as associating the body parts to identities, our key insight with regards to the pose formulation is to have *Stage I only consider body joints for which direct image evidence is available*, i.e., joints that are themselves visible or their kinematic parents are visible. This way *Stage I* does not have to spend representational capacity in hallucinating poses for joints that have no supporting image evidence. For each visible body joint, we predict the 2D part confidence maps, information for associating parts to an individual, and an *intermediate 3D pose encoding* for the bones that connect at the joint. Thus, the 3D pose encoding is only cognizant of the joint's immediate neighbours (*local*) in the kinematic chain. A compact fully-connected network forms *Stage II*, which relies on the intermediate pose encoding and other evidence extracted in the preceding stage, to decode the complete 3D pose. The *Stage II* network is able to reason about occluded joints using the full body context (*global*) for each detected subject, and leverages learned pose priors, and the subject 2D and 3D pose evidence. This stage is compact, highly efficient, and acts in parallel for all detected subjects.

Stage I is the most computationally expensive part of our pipeline, and the main bottleneck in achieving real-time performance.

We achieve real-time performance by contributing a new convolutional neural network (CNN) architecture in *Stage I* to speed up the most computationally expensive part of our pipeline. We refer to the new architecture as *SelecSLS Net.* Our proposed architecture depends on far fewer features than competing ones, such as ResNet-50 [He et al. 2016], without any accuracy loss thanks to our insights on selective use of short and long range concatenation-skip connections. This enables fast inference on the complete input frame, without the added pre- or post-processing complexity of a separate bounding box tracker for each subject. Further, the compactness of our *Stage II* network, which reconciles the partially incomplete 2D pose and 3D pose encoding to a full body pose estimate, enables it to simultaneously handle many people with minimal overhead on top of *Stage I.* Additionally, we fit a model-based skeleton to the 3D and 2D predictions in order to satisfy kinematic constraints and reconcile the 2D and 3D predictions across time. This produces temporally stable predictions, with skeletal angle estimates, which can readily drive virtual characters.

In summary, our technical innovations and new design insights at the individual stages, as well as our insights guiding the proposed multi-stage design enable our final contribution: a complete algorithm for multi-people 3D motion capture from a single camera that achieves real-time performance without sacrificing reliability or accuracy. The run time of our system only mildly depends on the number of subjects in the scene, and even crowded scenes can be tracked at high frame rates. We demonstrate our system's performance on a variety of challenging multi-person scenes.

Project Website: http://gvv.mpi-inf.mpg.de/projects/XNect/

2 RELATED WORK

We focus our discussion on relevant 2D and 3D human pose estimation from monocular RGB methods, in both single- and multi-person scenarios–for overview articles refer to Sarafianos et al. [2016]; Xia et al. [2017]. We also discuss prior datasets, and neural network architectures that inspired ours.

Multi-Person 2D Pose Estimation: Multi-person 2D pose estimation methods can be divided into bottom-up and top-down approaches. Top-down approaches first detect individuals in a scene and fall back to single-person 2D pose approaches or variants for pose estimation [Gkioxari et al. 2014; Iqbal and Gall 2016; Papandreou et al. 2017; Pishchulin et al. 2012; Sun and Savarese 2011]. Reliable detection of individuals under significant occlusion, and tracking of people through occlusions remains challenging.

Bottom-up approaches instead first localize the body parts of all subjects and associate them to individuals in a second step. Associations can be obtained by predicting joint locations and their identity embeddings together [Newell and Deng 2017], or by solving a graph cut problem [Insafutdinov et al. 2017; Pishchulin et al. 2016]. This involves solving an NP-hard integer linear program which easily takes hours per image. The work of Insafutdinov et al. [2017] improves over Pishchulin et al. [2016] by including image-based pairwise terms and stronger detectors based on ResNet [He et al. 2016]. This way reconstruction time reduces to several minutes per frame. Cao et al. [2017] predict joint locations and part affinities (PAFs), which are 2D vectors linking each joint to its parent. PAFs allow quick and greedy part association, enabling real time mutli-person 2D pose estimation. Our *Stage I* uses similar ideas to localize and assign joints in 2D, but we also predict an intermediate 3D pose encoding per joint which enables our subsequent stage to produce accurate 3D body pose estimates.Güler et al. [2018] compute dense correspondences from pixels to the surface of SMPL [2015], but they do not estimate 3D pose.

Single-Person 3D Pose Estimation: Monocular single person 3D pose estimation was previously approached with generative methods using physics priors [Wei and Chai 2010], or semi-automatic analysis-by-synthesis fitting of parametric body models [Guan et al. 2009; Jain et al. 2010]. Recently, methods employing CNN based learning approaches led to important progress [Ionescu et al. 2014; Li and Chan 2014; Li et al. 2015; Pavlakos et al. 2017; Sigal et al. 2010; Sun et al. 2017, 2018; Tekin et al. 2016]. These methods can broadly be classified into direct regression and 'lifting' based approaches. Regressing straight from the image requires large amounts of 3D-pose labelled images, which are difficult to obtain. Therefore, existing datasets are captured in studio scenarios with limited pose and appearance diversity [Ionescu et al. 2014], or combine real and synthetic imagery [Chen et al. 2016]. Consequently, to address the 3D data scarcity, transfer learning using features learned on 2D pose datasets has been applied to improve 3D pose estimation [Mehta et al. 2017a,b; Popa et al. 2017; Sun et al. 2017; Tekin et al. 2017; Zhou et al. 2017].

'Lifting' based approaches predict the 3D pose from a separately detected 2D pose [Martinez et al. 2017]. This has the advantages that 2D pose datasets are easier to obtain in natural environments, and the lifting can be learned from MoCap data without overfitting on the studio conditions. While this establishes a surprisingly strong baseline, lifting is ill-posed and body-part depth disambiguation is often not possible without additional cues from the image. Other work has proposed to augment the 2D pose with relative depth ordering of body joints as additional context to disambiguate 2D to 3D lifting [Pavlakos et al. 2018a; Pons-Moll et al. 2014]. Our approach can be seen as a hybrid of regression and lifting methods: An encoding of the 3D pose of the visible joints is regressed directly from the image (Stage I), with each joint only reasoning about its immediate kinematic neighbours (local context). This encoding, along with 2D joint detection confidences augments the 2D pose and is 'decoded' into a complete 3D body pose by *Stage II* reasoning about all body joints (global context).

Some recent methods integrate a 3D body model [Loper et al. 2015] within a network, and train using a mixture of 2D poses and 3D poses to predict 3D pose and shape from single images [Kanazawa et al. 2018; Omran et al. 2018; Pavlakos et al. 2018b; Tung et al. 2017]. Other approaches optimize a body model or a template [Habermann et al. 2019; Xu et al. 2018] to fit 2D poses or/and silhouettes [Alldieck et al. 2019, 2018a,b; Bhatnagar et al. 2019; Bogo et al. 2016; Guler and Kokkinos 2019; Kolotouros et al. 2019b,a; Lassner et al. 2017]. Very few are able to work in real time, and none of them handles multiple people.

Prior real-time 3D pose estimation approaches [Mehta et al. 2017b] designed for single-person scenarios fail in multi-person scenarios. Recent offline single-person approaches [Kanazawa et al. 2019] produce temporally coherent sequences of SMPL [2015] parameters, but work only for unoccluded single-person scenarios. In contrast, our proposed approach runs in real time for multi-person scenarios, and produces temporally coherent joint angle estimates at par with offline approaches, while successfully handling object and inter-person occlusions.

Multi-Person 3D Pose: Earlier work on monocular multi-person 3D pose capture often followed a generative formulation, e.g. estimating 3D body and camera pose from 2D landmarks using a learned pose space [Ramakrishna et al. 2012]. We draw inspiration from and improve over limitations of recent deep learning-based methods. Rogez et al. [2017] use a Faster-RCNN [2015] based approach and first find representative poses of discrete pose clusters that are subsequently refined. The LCRNet++ implementation of this algorithm uses a ResNet-50 base network and achieves non-real-time interactive 10 − 12fps on consumer hardware even with the faster but less accurate 'demo' version that uses fewer anchor poses. Dabral et al. [2019] use a similar Faster-RCNN based approach, and predict 2D keypoints for each subject. Subsequently, the predicted 2D keypoints are lifted to 3D pose. We show that incorporating additional information, such as the keypoint confidence, and 3D pose encodings in the 'lifting' step results in a much higher prediction accuracy. Moon et al. [2019] employ a prior person detection step, and pass resized image crops of each detected subject to the pose estimation network. As prior work [Cao et al. 2017] has shown, such an approach results high pose estimation accuracy, but comes at the cost of a significant increase in inference time. Not only does such an approach work at offline rates, the per-frame inference time scales with the number of subjects in the scene, making it unsuitable for real-time applications.

The aforementioned detection based approaches predict multiple proposals per individual and fuse them afterwards. This is time consuming, and in many cases it can either incorrectly merge nearby

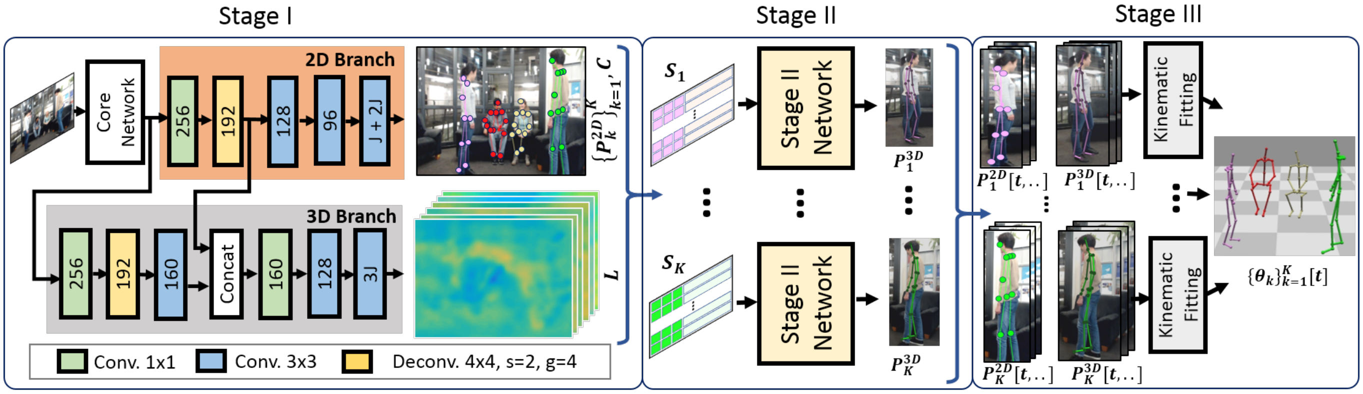

Fig. 2. **Overview:** Computation is separated into three stages, the first two respectively performing per-frame local (per body joint) and global (all body joints) reasoning, and the third performing temporal reasoning across frames: *Stage I* infers 2D pose and intermediate 3D pose encoding for visible body joints, using a new *SelecSLS Net* architecture. The 3D pose encoding for each joint only considers local context in the kinematic chain. *Stage II* is a compact fully-connected network that runs in parallel for each detected person, and reconstructs the complete 3D pose, including occluded joints, by leveraging global (full body) context. *Stage III* provides temporal stability, localization relative to the camera, and a joint angle parameterization through kinematic skeleton fitting.

individuals with similar poses, or fail to merge multiple proposals for the same individual. Beyond the cost and potential errors from fusing pose estimates, multiple detections of the same subject further increase the inference time for the approach of Moon et al. [2019].

Our approach, being bottom-up, does not produce multiple detections per subject. The bottom-up approach of Mehta et al. [2018b] predicts the 2D and 3D pose of all individuals in the scene using a fixed number of feature maps, which jointly encode for any number of individuals in the scene. This introduces potential *conflicts* when subjects overlap, for which a complex encoding and read-out scheme is introduced. The 3D encoding treats each limb and the torso as distinct objects, and encodes the 3D pose of each 'object' in the feature maps at the pixel locations corresponding to the 2D joints of the 'object'. The encoding can thus handle partial inter-personal occlusion by dissimilar body parts. Unfortunately, the approach still fails when similar body parts of different subjects overlap. Similarly, Zanfir et al. [2018b] jointly encode the 2D and 3D pose of all subjects in the scene using a fixed number of feature maps. Different from Mehta et al. [2018b], they encode the full 3D pose vector at all the projected pixels of the skeleton, and not just at the body joint locations, which makes the 3D feature space rife with potential encoding conflicts. For association, they learn a function to evaluate limb grouping proposals. A 3D pose decoding stage extracts 3D pose features per limb and uses an attention mechanism to combine these into a 3D pose prediction for the limb.

One of our key contributions is to only reason about body joints for which there is direct image evidence available, i.e., the joint itself, or its parent/child is visible. A subsequent compact fully-connected network can decode this potentially incomplete information to the full 3D pose. Such a hybrid of image-to-pose regression and 2D-to-3D lifting helps overcome the limitations of the individual approaches. The 3D pose encodings are a strong cue for the 3D pose in the absence of conflicts, whereas the global context in *Stage II* and the 2D pose help resolve the very-limited conflicts when they occur, and use learned pose priors to fill in the missing body joints. In contrast, the encodings of Zanfir et al. [2018b] and Mehta et al. [2018b] encode the pose for all joints in the full body or limb-wise, respectively, regardless of available image evidence for each joint, making the already difficult task more difficult, and having several avenues of potential encoding conflicts. Furthermore, we impose kinematic constraints with a model based fitting stage, which also allows for temporal smoothness. The approach of Zanfir et al. [2018a] also combines learning and optimization, but their space-time optimization over all frames is not real-time.

Different from prior approaches, our approach works in real-time at 25 – 30 fps using a single consumer GPU, yielding skeletal joint angles and camera relative positioning of the subject, which can be readily be used to control animated characters in a virtual environment. Our approach predicts the complete body pose even under significant person-object occlusions, and is more robust to inter-personal occlusions.

3D Pose Datasets: There exist many datasets with 3D pose annotations in single-person scenarios [Ionescu et al. 2014; Mehta et al. 2017a; Sigal et al. 2010; Trumble et al. 2017; von Marcard et al. 2016] or multi-person with only 2D pose annotations [Andriluka et al. 2014; Lin et al. 2014]. As multi-person 3D pose estimation started to receive more attention, datasets such as MarCOnI [Elhayek et al. 2016] with a lower number of scenes and subjects, and the more diverse Panoptic [Hanbyul Joo and Sheikh 2015] and MuCo-3DHP [Mehta et al. 2018b] datasets have come about. LCRNet [Rogez et al. 2017] uses 2D to 3D lifting to create pseudo annotations on the MPII 2D pose dataset [Andriluka et al. 2014], and LCRNet++ [Rogez et al. 2019] uses synthetic renderings of humans from a multitude of single person datasets.

Recently, the 3D Poses in the Wild (3DPW) dataset [von Marcard et al. 2018] features multiple people outdoors recorded with a moving camera and includes ground truth 3D pose. The number of subjects with ground truth pose is however limited. To obtain more variation in training, we use the recently published MuCo-3DHP [Mehta et al. 2018b], which is a multi-person training set of composited real images with 3D pose annotations from the single person MPI-INF-3DHP [2017a] dataset.

Convolutional Network Designs: ResNet [He et al. 2016] and derivatives [Xie et al. 2017] incorporate explicit information flowing from earlier to later feature layers in the network through summation-skip connections. This permits training of deeper and more powerful networks. Many architectures based on this concept have been proposed, such as Inception [Szegedy et al. 2017] and ResNext [Xie et al. 2017].

Because increased depth and performance comes at the price of higher computation times during inference, as well as a higher number of parameters, specialized architectures for faster test time computation were proposed, such as AmoebaNet [Real et al. 2019], MobileNet [Howard et al. 2019; Sandler et al. 2018], ESPNet [Mehta et al. 2018a], ERFNet [Romera et al. 2018], EfficientNet [Tan and Le 2019], as well as architectures such as SqueezeNet[Iandola et al. 2016] that target parameter efficiency. These are however not suited for our use case for various reasons: Many architectures with depthwise convolutions are optimized for inference on specific edge devices [Sandler et al. 2018], and lose accuracy in lieu of speed. Increasing the width or depth of these networks [Howard et al. 2019; Tan and Le 2019] to bring the accuracy closer to that of vanilla ResNets results in GPU runtimes comparable to typical ResNet architectures. ESPNet uses hierarchical feature fusion but produces non-smooth output maps with grid artifacts due to the use of dilated convolutions. These artifacts impair part association performance in our pose estimation setting. ShuffleNet [Ma et al. 2018; Zhang et al. 2018] use group convolutions and depthwise convolutions, and shuffle the channels between layers to promote information flow between channel groups. DenseNet [Huang et al. 2017b] uses full dense concatenation-skip connectivity, which results in a parameter efficient network but is slow due to the associated memory cost of the enormous number of concatenation operations. Recent work has also proposed highly computation-efficient networks [Sun et al. 2019a; Wang et al. 2019] which maintain high to low resolution feature representations throughout the network, and do not lose accuracy in lieu of computational efficiency. However, the theoretical computational efficiency does not translate to computational speed in practice, and the models are up to twice as slow as ResNet networks for a given accuracy level. Approaches targeting parameter efficiency often do not result in computational speedups because either the computational cost is still high [Iandola et al. 2016], or the non-structured sparse operations resulting from weight pruning [Frankle and Carbin 2018] cannot be executed efficiently on current hardware.

The key novel insight behind our proposed CNN architecture is the use of selective long-range and short-range concatenation-skip connections rather than the dense connectivity pattern of DenseNet. This results in a network significantly faster than ResNet-50 while retaining the same level of accuracy, avoids the artifacts and accuracy deficit of ESPNet, and eliminates the memory (and hence speed) bottlenecks associated with DenseNet.

3 METHOD OVERVIEW

The input to our method is a live video feed, i.e., a stream of monocular color frames showing a multi-person scene. Our method has three subsequent stages, as shown in Fig. 2. In Section 4, we discuss the first two stages, which together produce 2D and 3D pose estimates per frame.

Stage I uses a convolutional neural network to process the complete input frame, jointly handling all subjects in the scene. The *Stage I* CNN predicts 2D body joint heatmaps, Part Affinity Fields to associate joints to individuals in the scene, and an intermediate 3D pose encoding per detected joint. After grouping the 2D joint detections from the first stage into individuals following the approach of [Cao et al. 2017], 3D pose encodings per individual are extracted at the pixel locations of the visible joints and are input to the second stage together with the 2D locations and detection confidences of the individual's joints. *Stage I* only reasons about visible body joints, and the 3D pose encoding per joint only captures the joint's pose relative to its immediate kinematic neighbours. The 3D pose encoding is discussed in Section 4.1.2.

Stage II, which we discuss in Section 4.2, uses a lightweight fully-connected neural network that 'decodes' the input from the previous stage into a full 3D pose, i.e. root-relative 3D joint positions for visible and occluded joints, per individual. This network incorporates 2D pose and 3D pose encoding evidence over all visible joints and an implicitly learned prior on 3D pose structure, which allows it to reason about occluded joints and correct any 3D pose encoding conflicts. A further advantage of a separate stage for full 3D pose reasoning is that it allows the use of a body joint set different from that used for training *Stage I*. In our system, 3D pose inference of *Stage I* and *Stage II* can be parallelized on a GPU, with negligible dependence of inference time on the number of subjects.

Stage III, discussed in Section 5, performs sequential model fitting on the live stream of 2D and 3D predictions from the previous stages. A kinematic skeleton is fit to the history of per-frame 2D and root-relative 3D pose predictions to obtain temporally coherent motion capture results. We also track person identity, full skeletal joint angles, and the camera relative localization of each subject in real time.

Our algorithm and pose representation applies to any CNN architecture suitable for keypoint prediction. However, to enable fast inference on typical consumer-grade GPUs, we propose the novel *SelecSLS Net* architecture for the backbone of *Stage I* CNN. It employs selective long and short range concatenation-skip connections to promote information flow across network layers which allows to use fewer features and have a much smaller memory footprint leading to a much faster inference time but comparable accuracy in comparison to ResNet-50. We discuss our contributions in that regard separately in Section 6.

In Section 7, we present ablation and comparison studies, both quantitative and qualitative, and show applications to animated character control.

4 PER-FRAME POSE ESTIMATION: *STAGE I* & *STAGE II*

Given an image I of dimensions $w \times h$ pixels, we seek to estimate the 3D pose $\{P_k^{3D}\}_{k=1}^{K}$ of the unknown number of K individuals in the scene. $P_k^{3D} \in \mathbb{R}^{3 \times J}$ represents the root (pelvis)-relative 3D coordinates of the J body joints. The task is implemented in the first two stages of our algorithm, which we detail in the following.

4.1 *Stage I* Prediction

Our first stage uses a CNN that features an initial core (or backbone) network that splits into two separate branches for 2D pose prediction and 3D pose encoding, as shown in Figure 2. The core network outputs

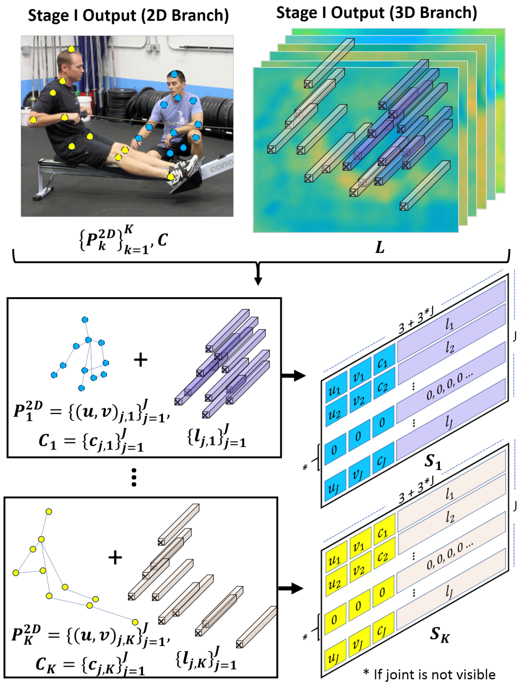
Stage I Output (2D Branch)
Stage I Output (3D Branch)
$\{P_k^{2D}\}_{k=1}^{K}, C$
L
+
$P_1^{2D} = \{(u,v)_{j,1}\}_{j=1}^{J}$,
$C_1 = \{c_{j,1}\}_{j=1}^{J}$
$\{l_{j,1}\}_{j=1}^{J}$
+
$P_K^{2D} = \{(u,v)_{j,K}\}_{j=1}^{J}$,
$C_K = \{c_{j,K}\}_{j=1}^{J}$
$\{l_{j,K}\}_{j=1}^{J}$
3 + 3*J
J
u_1 v_1 c_1
u_2 v_2 c_2
0 0 0
u_J v_J c_J
l_1
l_2
0, 0, 0, 0 ...
l_J
*
S_1
S_K
* If joint is not visible

Fig. 3. **Input to *Stage II***: S_k for each detected individual k, is comprised of the individual's 2D joint locations P_k^{2D}, the associated joint detection confidence values C extracted from the 2D branch output, and the respective 3D pose encodings $\{l_{j,k}\}_{j=1}^{J}$ extracted from the output of the 3D branch. Refer to Section 4 for details.

features at $\frac{w}{16} \times \frac{h}{16}$ pixel spatial resolution, and uses our new proposed network design that offers a high accuracy at high runtime efficiency, which we detail in Section 6. The outputs of each of the 2D and 3D branches are at $\frac{w}{8} \times \frac{h}{8}$ pixels spatial resolution. The 3D pose branch also makes use of features from the 2D pose branch. We explain both branches and the *Stage I* network training in the following.

4.1.1 2D Branch: 2D Pose Prediction and Part Association. 2D pose is predicted as 2D heatmaps $H = \{H_j \in \mathbb{R}^{\frac{w}{8} \times \frac{h}{8}}\}_{j=1}^{J}$, where each map represents the per-pixel confidence of the presence of body joint type j jointly for all subjects in the scene. Similar to [Cao et al. 2017], we use Part Affinity fields $F = \{F_j \in \mathbb{R}^{\frac{w}{8} \times \frac{h}{8} \times 2}\}_{j=1}^{J}$ to encode body joint ownership using a unit vector field that points from a joint to its kinematic parent, and spans the width of the respective limb. For an input image, these Part Affinity Fields can be used to detect the individuals present in the scene and the visible body joints, and to associate visible joints to individuals. If the neck joint (which we hypothesize is visible in most situations) of an individual is not detected, we discard that individual entirely from the subsequent stages. For K detected individuals, this stage outputs the 2D body joint locations in absolute image coordinates $P_k^{2D} \in \mathbb{Z}_+^{2 \times J}$. Further, we get an estimate of the detection confidence $c_{j,k}$ of each body part j and person k from the heatmap maximum.

4.1.2 3D Branch: Predicting Intermediate 3D Pose Encoding. The 3D branch of the *Stage I* network uses the features from the core network and the 2D branch to predict 3D pose encoding maps $L = \{L_j \in R^{\frac{w}{8} \times \frac{h}{8} \times 3}\}_{j=1}^{J}$. The encoding at the spatial location of each visible joint only encapsulates its 3D pose relative to the joints to which it directly connects in the kinematic chain.

The general idea of such an encoding map is inspired by the approaches of Mehta et al. [2017b], Pavlakos et al. [2017] which represent the 3D pose information of joints in output maps at the spatial locations of the 2D detections of the respective joints.

Our specific encoding in L works as follows: Consider the $1 \times 1 \times (3 \cdot J)$ vector $l_{j,k}$ extracted at the pixel location $(u,v)_{j,k}$ from the 3D output maps L. Here $(u,v)_{j,k}$ is the location of body joint j of individual k. This $1 \times 1 \times (3 \cdot J)$ feature vector is of the dimensions of the full 3D body pose, where the kinematic parent-relative 3D locations of each joint reside in separate channels. Importantly however, and in contrast to previous work [Mehta et al. 2018b; Zanfir et al. 2018b], instead of encoding the full 3D body pose, or per-limb pose, at each 2D detection location $(u,v)_{j,k}$, we only encode the pose of the corresponding joint (relative to its parent) and the pose of its children (relative to itself). In other words, at each joint location $(u,v)_{j,k}$, we restrict the supervision of the encoding vector $l_{j,k}$ to the subset of channels corresponding to the bones that meet at joint j, parent-to-joint and joint-to-child in the kinematic chain. We will refer to this as channel-sparse supervision of $\{l_{j,k}\}_{j=1}^{J}$, and emphasize the distinction from channel-dense supervision. Figure 4 shows examples for head, neck and right shoulder. Consequently, 3D pose information for all the visible joints of all subjects is still encoded in L, albeit in a spatially distributed manner, and each 2D joint location $(u,v)_{j,k}$ is used to extract its corresponding 3D bones of subject k. Our motivation for such a pose encoding is that the task of parsing in-the-wild images to detect 2D body part heatmaps under occlusion and clutter, as well as grouping the body parts with their respective person identities under inter-personal interaction and overlap is already challenging. Reasoning about the full 3D pose, including occluded body parts, adds further complexity, which not only requires increased representation capacity (thus increasing the inference cost), but also more labelled training data, which is scarce for multi-person 3D pose. The design of our formulation responds to both of these challenges. Supervising only the 3D bones corresponding to each visible joint ensures that mostly local image evidence is used for prediction, where the full body context is already captured by the detected 2D pose. For instance, it should be possible to infer the kinematic-parent relative pose of the upper arm and the fore arm by looking at the region centered at the elbow. This means better generalization and less risk to overfit to dataset specific long-range correlations.

Further, our use of channel-sparse (joint-type-dependent) supervision of $l_{j,k}$ is motivated by the fact that convolutional feature maps cannot contain sharp transitions [Mehta et al. 2018b]. In consequence, full poses of two immediately nearby people are hard to encode. E.g., the wrist of one person being in close proximity in the image plane to the shoulder of another person would require the full pose of two different individuals to be encoded in possibly adjacent pixel locations in the output map. Such encoding conflicts often lead to failures of previous methods, as shown in Figure 5 in the supplemental

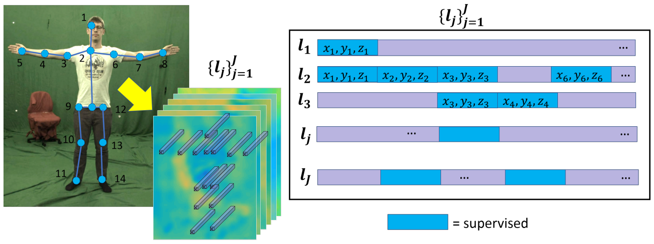

Fig. 4. **3D Pose Encoding With Local Kinematic Context:** The supervision for the $1 \times 1 \times (3 \cdot J)$ predicted 3D pose encoding vector l_j (shown on the right) at each joint j (shown on the skeleton on the left) is dependent on the type of the joint. l_j only encodes the 3D pose information of joint j relative to the joints it is directly connected to in the kinematic chain. This results in a channel-sparse supervision pattern as shown here, as opposed to each l_j encoding the full body pose. The regions marked purple are not supervised and the network is free to predict any values there. For example, 'right shoulder' ($j = 3$) connects to joints $j = 2$ and $j = 4$, so in l_3 the pose of $j = 3$ relative to $j = 2$ (x_3, y_3, z_3), and the pose of $j = 4$ relative to $j = 3$ (x_4, y_4, z_4) are supervised. See Section 4.1.2.

document. In contrast, our encoding in L does not lead to encoding conflicts when different joints of separate individuals are in spatial proximity or even overlap in the image plane, because supervision is restricted to the channels corresponding to the body joint type. Consequently, our target output maps are smoother without sharp transitions, and more suitable for representation by CNN outpus. In Section 7.5 we show the efficacy of channel-sparse supervision for $\{l_{j,k}\}_{j=1}^J$ over channel-dense supervision across various 2D and 3D pose benchmarks. Importantly, unlike Zanfir et al. [2018b] and Mehta et al. [2018b], the 2D pose information is not discarded, and is utilized as additional relevant information for 3D pose inference in *Stage II*, allowing for a compact and fast network. This makes it more suited for a real-time system than, for instance, the attention-mechanism-based inference scheme of Zanfir et al. [2018b].

For each individual k, the 2D pose P_k^{2D}, joint confidences $\{c_{j,k}\}_{j=1}^J$, and 3D pose encodings $\{l_{j,k}\}_{j=1}^J$ at the visible joints are extracted and input to Stage II (Sec. 4.2). Stage II uses a fully-connected decoding network that leverages the full body context that is available to it, to give the complete 3D pose with the occluding joints filled in. We provide details of Stage II in Section 4.2.

4.1.3 Stage I Training. The *Stage I* network is trained in multiple stages. First the core network and the 2D pose branch are trained for single person 2D pose estimation on the MPII [Andriluka et al. 2014] and LSP [Johnson and Everingham 2010, 2011] single person 2D datasets. Then, using these weights as initialization, it is trained for multi-person 2D pose estimation on MS-COCO [Lin et al. 2014]. Subsequently the 3D pose branch is added and the two branches are individually trained on crops from MS-COCO and MuCo-3DHP [Mehta et al. 2018b], with the core network seeing gradients from both datasets via the two branches. Additionally, the 2D pose branch sees supervision from MuCo-3DHP dataset via heatmaps of the common minimum joint set between MS-COCO and MuCo-3DHP. We found that the pretraining on multi-person 2D pose data before introducing the 3D branch is important.

4.2 *Stage II* Prediction

Stage II uses a lightweight fully-connected network to predict the root-relative 3D joint positions $\{P_k^{3D}\}_{k=1}^K$ for each individual considered visible after *Stage I*. Before feeding the output from *Stage I* as input, we convert the 2D joint position predictions P_k^{2D} to a representation relative to the neck joint. For each individual k, at each detected joint location, we extract the $1 \times 1 \times (3 \cdot J)$ 3D pose encoding vector $l_{j,k}$, as explained in the preceding section. The input to *Stage II*, $S_k \in \mathbb{R}^{J \times (3+3 \cdot J)}$, is the concatenation of the neck relative $(u,v)_{j,k}$ coordinates of the joint, the joint detection confidence $c_{j,k}$ and the feature vector $l_{j,k}$, for each joint j. If the joint is not visible, we instead concatenate zero vectors of appropriate dimensions (see Figure 3). *Stage II* comprises a 5-layer fully-connected network, which converts S_k to a root-relative 3D pose estimate P_k^{3D} (see Figure 5).

We emphasize that unlike Mehta et al. [2018b], we have a much sparser pose encoding $l_{j,k}$, and further only use it as a feature vector and not directly as the body part's pose estimate because jointly encoding body parts of all individuals in the same feature volume may result in corrupted predictions in the case of conflicts–same parts of different individuals in close proximity. Providing the 2D joint positions and part confidences along with the feature vectors as input to the *Stage II* network allows it to correct any conflicts that may arise. See Figure 5 in the supplemental document for a visual comparison of results against [Mehta et al. 2018b].

The inference time for *Stage II* with a batch size of 10 is 1.6ms on an Nvidia K80, and 1.1ms on a TitanX (Pascal).

4.2.1 Stage II Training. The *Stage II* network is trained on uncropped frames from MuCo-3DHP [Mehta et al. 2018b]. We run *Stage I* on these frames and extract the 2D pose and 3D pose encodings. Then for each detected individual, we use the ground-truth root-relative 3D pose as the supervision target for $\{(X_j, Y_j, Z_j)\}_{j=1}^J$. Since the pose prediction can be drastically different from the ground truth when there are severe occlusions, we use the smooth-L1 [Ren et al. 2015] loss to mitigate the effect of such outliers. In addition to providing an opportunity to reconcile the 3D pose predictions with the 2D pose, another advantage of a second stage trained separately from the first stage is that the output joint set can be made different from the joint set used for *Stage I*, depending on which dataset was used for training *Stage II* (joint sets typically differ across datasets). In our case, though there are no 2D predictions for foot tip, the 3D pose encoding for ankle encodes information about the foot tip, which is used in *Stage II* to produce 3D predictions for foot tips.

5 SEQUENTIAL MOTION CAPTURE: *STAGE III*

After *Stage I* and *Stage II* we have per-frame root-relative pose estimates for each individual. However, we have no estimates of person size or metric distance from the camera, person identities are not tracked across frames, and reconstructions are not in terms of joint angles. To remedy this, we infer and track person appearance over time, optionally infer absolute height from ground plane geometry, and fuse 2D and 3D predictions with temporal smoothness and joint limit constraints in a space-time kinematic pose fitting method.

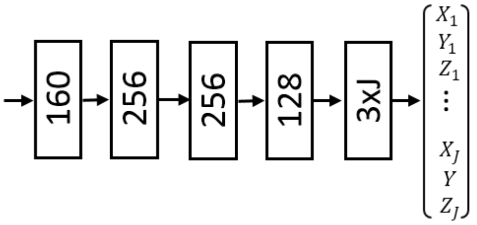

Fig. 5. Lightweight fully connected network that forms *Stage II* of our pipeline. The network decodes the inferred 2D body pose, joint detection confidences and 3D pose encodings coming from *Stage I* to root-relative full body 3D pose (X_j, Y_j, Z_j) estimates, leveraging full body context to fill in occluded joints.

5.1 Identity Tracking and Re-identification

To distinguish poses estimated at distinct frames, we extend the previous pose notation with temporal indices in square brackets. So far, per-frame 2D and 3D poses have been estimated for the current and past frames. We need a fast method that maintains identity of a detected person across frames and re-identifies it after a period of full occlusion. To this end, we assign correspondences between person detections at the current timestep t, $\{P_i[t]\}_{i=1}^{K[t]}$, to the preceding ones $\{P_k[t-1]\}_{k=1}^{K[t-1]}$. We model and keep track of person appearance with an HSV color histogram of the upper body region. We discretize the hue and saturation channels into 30 bins each and determine the appearance $A_{i[t]}$ as the class probabilities across the bounding box enclosing the torso joints in $\{P_i^{2D}[t]\}_i$. This descriptor is efficient to compute and can model loose and tight clothing alike, but might suffer from color ambiguities across similarly dressed subjects. Other, more complex identity tracking methods can be used instead when real-time performance is not needed, such as when processing pre-recorded sequences.

To be able to match subjects robustly, we assign current detections to previously known identities not only based on appearance similarity, $S_{i,k}^{A} = (A_i[t] - A_k[t-1])^2$, but also on the 2D pose similarity $S_{i,k}^{P2D}(i,k) = (P_{i[t]}^{2D} - P_{k[t-1]}^{2D})^2$ and 3D pose similarity $S_{i,k}^{P3D}(i,k) = (P_{i[t]}^{3D} - P_{k[t-1]}^{3D})^2$. A threshold on the dissimilarity is set to detect occlusions, persons leaving the field of view, and new persons entering. That means the number of persons $K[t]$ can change. Person identities are maintained for a certain number of frames after disappearance, to allow for re-identification after momentary occlusions such as those caused by the tracked subjects passing behind an occluder. We update the appearance histogram of known subjects at arrival time and every 30 seconds to account for appearance changes such as varying illumination.

5.2 Relative Bone Length and Absolute Height Calculation

Relative bone length between body parts is a scale-invariant property that is readily estimated by P_k^{3D} in *Stage II*. To increase robustness, we take the normalized skeleton bone lengths b_k as the distance between linked joints in P_k^{3D} averaged across the first 10 frames.

Translating relative pose estimates from pixel coordinates to absolute 3D coordinates in cm is a difficult task as it requires either a reference object of known position and scale or knowledge of the person's height, which in turn can only be guessed with uncertainty from monocular footage [Gunel et al. 2019].

In Section 5.3 we explain how the camera relative position up to a scale is recovered through a re-projection constraint. We can optionally utilize the ground plane as reference geometry since camera calibration is less cumbersome than measuring the height of every person appearing in the scene. See the supplementary document for details.

5.3 Kinematic Skeleton Fitting

After 2D and 3D joint position prediction, we optimize for the skeletal pose $\{\theta_k[t]\}_{k=1}^{K[t]}$ of all $K[t]$ people in the scene, with $\theta_k[t] \in \mathbb{R}^D$ where $D = 29$ is the number of degrees of freedom (DOF) for one skeleton. Both, per-frame 2D and 3D pose estimates from previous stages are temporally filtered [Casiez et al. 2012] before skeleton fitting. Note that $\theta_k \in \mathbb{R}^D$ describes the pose of a person in terms of joint angles of a fixed skeleton plus the global root position, meaning our final output is directly compatible with CG character animation pipelines. We jointly fit to both 2D and root-relative 3D predictions as this leads to better reprojection error while maintaining plausible and robust 3D articulation. We estimate $\theta_k[t]$ by minimizing the fitting energy

$$\begin{aligned}\mathcal{E}(\theta_1[t],\cdots,\theta_K[t]) = w_{\text{3D}}E_{\text{3D}} + w_{\text{2D}}E_{\text{2D}} + w_{\text{lim}}E_{\text{lim}} \\ + w_{\text{temp}}E_{\text{temp}} + w_{\text{depth}}E_{\text{depth}}\,. \qquad (1)\end{aligned}$$

We formulate $\frac{\partial \mathcal{E}}{\partial \theta_k[t]}$ in closed form to perform efficient minimization by gradient descent using a custom implementation. The influence of the individual terms is balanced with w_{3D} = 9e−1, w_{2D} = 1e−5, w_{lim} = 5e−1, w_{temp} = 1e−7, and w_{depth} = 8e−6. In the following, we explain each term in more detail.

3D Inverse Kinematics Term: The 3D fitting term measures the 3D distance between predicted root-relative 3D joint positions $P_k^{3D}[t]$ and the root-relative joint positions in the skeleton $\bar{\mathcal{P}}(\theta_k[t],b_k)$ posed by forward kinematics for every person k, joint j and previously estimated relative bone lengths b_k,

$$E_{\text{3D}} = \sum_{k=1}^{K}\sum_{j=1}^{J3D} ||\bar{\mathcal{P}}(\theta_k[t],b_k)_j - P_{k,j}^{3D}[t]||_2^2 \quad . \qquad (2)$$

2D Re-projection Term: The 2D fitting term is calculated as the 2D distance between predicted 2D joint positions $P_k^{2D}[t]$ and the projection of the skeleton joint positions $\mathcal{P}(\theta_k[t],b_k)_j$ for every person k and joint j,

$$E_{\text{2D}} = \sum_{k=1}^{K}\sum_{j=1}^{J2D} w_j^{2D} c_{j,k} ||\Pi(h_k \mathcal{P}(\theta_k[t],b_k))_j - P_{k,j}^{2D}[t]||_2^2 \quad , \qquad (3)$$

where c is the 2D prediction confidence, w_j^{2D} is per-joint relative weighting, and Π is the camera projection matrix. The lower limb joints have a relative weighting of 1.7, elbows 1.5 and wrist joints 2.0 as compared to torso joints (hips, neck, shoulders). Note that $\mathcal{P}$ outputs unit height, the scaling with h_k maps it to metric coordinates, and the projection constraint thereby reconstructs absolute position in world coordinates.

Joint Angle Limit Term: The joint limits regularizer enforces a soft limit on the amount of joint angle rotation based on the anatomical joint rotation limits θ^{min} and θ^{max}. We write it as

$$E_{\text{lim}} = \sum_{k=1}^{K}\sum_{j=7}^{D} \begin{cases} (\theta_j^{min} - \theta_{k,j}[t])^2 & ,\text{if } \theta_{k,j}[t] < \theta_j^{min} \\ (\theta_{k,j}[t] - \theta_j^{max})^2 & ,\text{if } \theta_{k,j}[t] > \theta_j^{max} \\ 0 & ,\text{otherwise} \end{cases} \quad , \qquad (4)$$

where we start from $j = 7$ since we do not have limits on the global position and rotation parameters. Note that our neural network is trained to estimate joint positions and hence has no explicit knowledge about joint angle limits. Therefore, E_{lim} ensures biomechanical plausibility of our results.

Temporal Smoothness Term: Since our neural network estimates poses on a per-frame basis, the results might exhibit temporal jitter. The temporal stability of our estimated poses is improved by

$$E_{\text{temp}}(\Theta) = \sum_{k=1}^{K} ||\nabla\theta_k[t-1] - \nabla\theta_k[t]||_2^2 \quad , \qquad (5)$$

where the rate of change in parameter values, $\nabla\theta_k$, is approximated using backward differences. In addition, we penalize variations in the less constrained depth direction stronger, using the smoothness term $E_{\text{depth}} = ||\theta_{k,2}[t]_z - \theta_{k,2}[t-1]_z||$, where $\theta_{k,2}$ is the degree of freedom that drives the z-component of the root position.

Inverse Kinematics Tracking Initialization: For the first frame of a new person track, the local joint angles of the skeleton are fit to the 3D prediction only considering E_{3D} and E_{lim}. The local angles are then held fixed while minimizing E_{2D} to find the best fit for global translation and rotation of the skeleton. Subsequently, the complete energy formulation $\mathcal{E}(\theta_1[t], \cdots, \theta_K[t])$ is used.

6 SELECSLS NET: A FAST AND ACCURATE POSE INFERENCE CNN

Our *Stage I* core network is the most expensive component of our algorithm in terms of computation time. We evaluate various popular network architectures (see Table 1 in the supplemental document) on the task of single person 2D pose estimation, and determine that despite various parameter-efficient depthwise-convolution-based designs, for GPU-based deployment, ResNet architectures provide a better or comparable speed–accuracy tradeoff, and thus we use be them as *baselines*.

ResNet-50 [2016] has been employed for other multi-person pose estimation methods such as Mehtaet al. [2018b], Rogez et al. [2019], and Dabral et al. [2019]. Only on top-end hardware (Nvidia 1080TI, 11.4TFLOPs) does the ResNet-50 variant of our system run in real time. For more widespread lower performance hardware like ours (Nvidia 1080-MaxQ, 7.5TFLOPs), ResNet-50-based *Stage I* forward pass takes $>$30ms, which, together with the additional overhead of $\approx$ 15ms from the other components of our system, does not reach real-time performance of $>$25 fps.

We therefore propose a new network architecture module, called *SelecSLS* module, that uses short range and long range concatenation-skip connections in a selective way instead of additive-skip connections. SelectSLS is the main building block of the novel *SelecSLS Net* architecture for the *Stage I* core CNN. Additive-skip, as is used in ResNet architectures, performs element-wise addition to the features at the skip connection point, whereas concatenative-skip connections, as used in DenseNet, performs concatenation along the channel-dimension. Our new selective use of concatenation-skip connectivity promotes information flow through the network, without the exorbitant memory and compute cost of the full connectivity of DenseNet. Our new *Stage I* network shows comparable accuracy to a ResNet-50 core with a substantially faster inference time ($\approx$1.4×), across single person and multi-person 2D and 3D pose benchmarks.

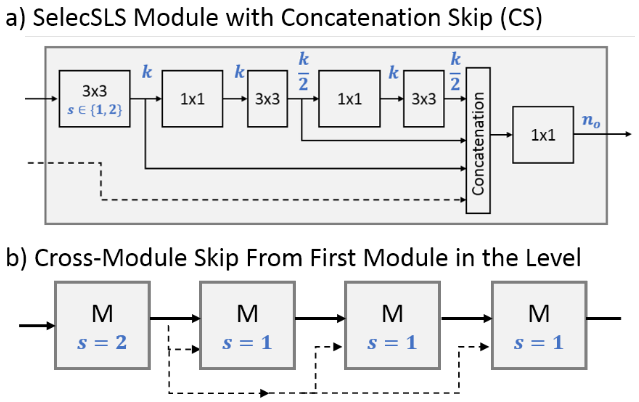
a) SelecSLS Module with Concatenation Skip (CS)
3x3
s ∈ {1, 2}
k
1x1
k
3x3
k/2
1x1
k
3x3
k/2
Concatenation
1x1
n_o
b) Cross-Module Skip From First Module in the Level
M
s = 2
M
s = 1
M
s = 1
M
s = 1

Fig. 6. The proposed *SelecSLS* module design (a) is comprised of interleaved 1×1 and 3×3 convolutions, and handles cross-module skip connections internally as concatenative-skip connections. The cross module skip connections themselves come from the first module that outputs features at a particular spatial resolution (b). See the supplemental document for ablation studies on alternative skip connectivity choices, through which this design emerged. Our design is parameterized by module stride (s), the number of intermediate features (k), and the number of module ouputs n_o.

6.1 *SelecSLS* Module

Our *Stage I* core network is comprised of building blocks, *SelecSLS* modules, with intra-module short-range skip connectivity and cross-module longer-range skip connectivity.The module design is as shown in Figure 6 (a) and (b). It comprises of a series of 3×3 convolutions interleaved with 1×1 convolutions. This is to enable mixing of channels when grouped 3×3 convolutions are used. All convolutions are followed by batch normalization and ReLU non-linearity. The module hyperparameter k dictates the number of features output by the convolution layers within the module. The outputs of all 3×3 convolutions ($2k$) are concatenated and fed to a 1×1 convolution which produces n_o features. The first 3×3 filters in the module are convolved with a stride of 1 or 2, which dictate the feature resolution of the entire module. The cross-module skip connection is the second input to the module.

Different from the traditional inter-module skip connectivity pattern, which connects each module to the previous one, our design uses long range skip connectivity to the first module of each level (Figure 6 (b)). This is intended to promote information flow through the network on the forward and the backward pass, and performs better than the traditional inter-module skip connectivity in practice. We define a level as all modules in succession which output feature

Table 1. *SelecSLS Net* Architecture: The table shows the network levels, overall number of modules, number of intermediate features k, the number of outputs of modules n_o, and the spatial resolution of features of the network proposed in Section 6.

Level	Output Resolution	*Module* Type	Stride s	k	Cross-Module Skip Conn.	n_o
L0	w/2 x h/2	Conv. 3x3	2	-	-	32
L1	w/4 x h/4	SelecSLS	2	64	No	64
	w/4 x h/4	SelecSLS	1	64	First	$\underline{128}$
L2	w/8 x h/8	SelecSLS	2	128	No	128
	w/8 x h/8	SelecSLS	1	128	First	128
	w/8 x h/8	SelecSLS	1	128	First	$\underline{288}$
L3	w/16 x h/16	SelecSLS	2	288	No	288
	w/16 x h/16	SelecSLS	1	288	First	288
	w/16 x h/16	SelecSLS	1	288	First	288
	w/16 x h/16	SelecSLS	1	288	First	416

maps of a particular spatial resolution. See the supplemental document for a detailed design ablation study leading to the proposed module design and the overall architecture.

6.2 *SelecSLS Net* Architecture

Table 1 shows the overall architecture of the proposed *SelecSLS Net*, parameterized by the stride of the module (s), the intermediate features in the module (k), and number of outputs of the module (n_o). All 3×3 convolutions with more than 96 outputs use a group size of 2, and those with more than 192 outputs use a group size of 4.

Design Evaluation: We compare the proposed architecture against ResNet-50 and ResNet-34 architectures as core networks to establish appropriate baselines. We compare our proposed architecture's performance, both on 2D pose estimation and 3D pose estimation in Section 7.4. As we show in Tables 2, 5, and 8, the proposed architecture is $1.3-1.8\times$ faster than ResNet-50, while retaining the same accuracy.

Memory Footprint: The speed advantage of *SelecSLS Net* over ResNet-50 stems primarily from its significantly smaller memory footprint. With a mini-batch comprised of a single 512×320px image, *SelecSLS Net* occupies just 80% of the memory (activations and bookkeeping for backward pass) occupied by ResNet-50. For larger mini-batch sizes, such as a mini-batch size of 32, the memory occupancy of *SelecSLS Net* is 50% of that of ResNet-50. Beyond the associated speed advantage, the significantly smaller memory footprint of *SelecSLS Net* allows mini-batches of twice the size compared to ResNet-50, both for training and batched inference.

7 RESULTS

In this section we evaluate the results of our real-time multi-person motion capture solution qualitatively and quantitatively on various benchmarks, provide extensive comparison with prior work, and conduct a detailed ablative analysis of the different components of our system. To ensure that the reported results on 3D pose benchmarks are actually indicative of the deployed system's performance, there is no test-time augmentation applied for our quantitative and qualitative results. We do not use procrustes alignment to the ground truth, nor do we rely on ground truth camera relative localization of the subjects to generate or modify our predictions.

Table 2. Evaluation of 2D keypoint detections of the complete *Stage I* of our system (both 2D and 3D branches trained), with different core networks on a subset of validation frames of MS COCO dataset. Also reported are the forward pass timings of the first stage on different GPUs (K80, TitanX (Pascal)) for an input image of size 512 × 320 pixels. We also show the 2D pose accuracy when using channel-dense supervision of $\{l_{j,k}\}_{j=1}^{J}$ in the 3D branch in place of our proposed channel-sparse supervision (Section 4.1.2).

Core	FP Time							
Network	K80	TitanX	AP	$\mathrm{AP}_{0.5}$	$\mathrm{AP}_{0.75}$	AR	$\mathrm{AR}_{0.5}$	$\mathrm{AR}_{0.75}$
ResNet-34	29.0ms	6.5ms	45.0	72.0	46.1	49.9	74.4	51.6
ResNet-50	39.3ms	10.5ms	46.6	73.0	48.9	51.4	75.4	54.0
SelecSLS	28.6ms	7.4ms	**47.0**	**73.5**	**49.5**	**51.8**	75.6	**54.1**
3D Branch With Channel-Dense $\{l_{j,k}\}_{j=1}^{J}$ Supervision								
SelecSLS	28.6ms	7.4ms	46.8	**73.5**	49.0	51.5	**75.9**	53.8

For additional qualitative results, please refer to the accompanying video.

7.1 System Characteristics and Applications

First, we show that our system provides efficient and accurate 3D motion capture that is ready for live character animation and other interactive CG applications, rivaling depth-based solutions despite using only a single RGB video feed.

Real-time Performance: Our live system uses a standard webcam as input, and processes 512×320 pixel resolution input frames. For a scene with 10 subjects, the system running on a Desktop with an Intel Xeon E5 with 3.5 GHz and an Nvidia GTX 1080Ti is capable of processing input at > 30 fps, while on a laptop with an Intel i7-8780H and a 1080-MaxQ it runs at ≈ 27 fps. On the laptop, *Stage I* takes 21.5 ms, part association and feature extraction take 1 ms, *Stage II* takes 1 ms, and *Stage III* takes ≈ 9 ms (2.4 ms for identity matching to the previous frame, 6.8 ms for skeleton fitting).

We compare our timing against the faster but less accurate 'demo' version of LCRNet++ [2019], but compare our accuracy against the slower but more accurate version of LCRNet++. LCRNet++ demo system uses ResNet-50 with less post-processing overhead than the accuracy-benchmarked system, and we measured its forward pass time on a TitanX-Pascal GPU (11 TFLOPs) to be 16 ms, while on a K80 GPU (4.1 TFLOPs) it takes >100 ms. Our *SelecSLS*-based system takes only 14 ms on TitanX-Pascal and 35 ms on a K80 GPU while producing more accurate per-frame joint position predictions (*Stage II*) than the slower version of LCRNet++, as shown in Table 7. An additional CPU-bound overhead of ≈ 9 ms for *Stage III* results in temporally smooth joint-angle estimates which can readily be used to drive virtual characters. The accompanying video contains examples of our live setup running on the laptop. Note that throughout the manuscript we report the timings of the various stages of our system on a set of GPUs with FP32 performance representative of current low-end and high-end consumer GPUs ($\approx 4-12$TFLOPs).

Multi-Person Scenes and Occlusion Robustness: In Figure 9, we show qualitative results of our full system on several scenes containing

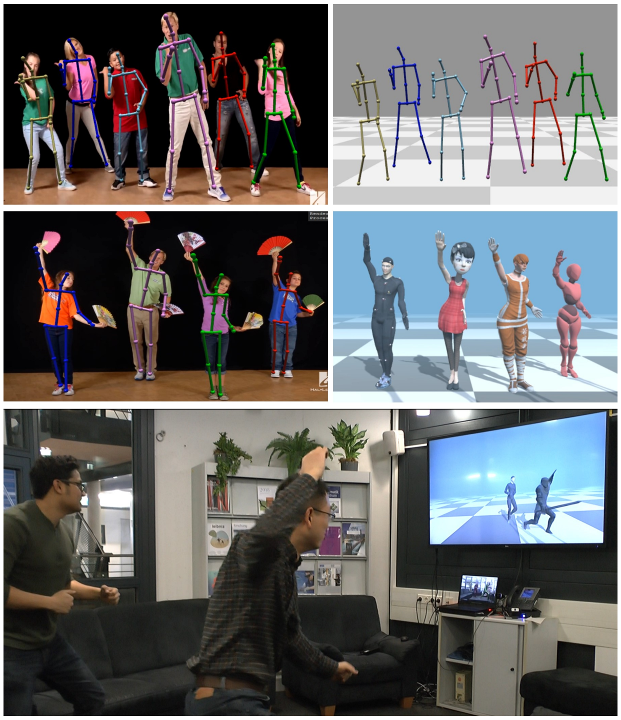

Fig. 7. **Live Interaction and Virtual Character Control:** The temporally smooth joint angle predictions from *Stage III* can be readily employed for driving virtual characters in real-time. The top two rows show our system driving virtual skeletons and characters with the motion captured in real time. On the bottom, our system is set up as a Kinect-like game controller, allowing subjects to interact with their virtual avatars live. Some images courtesy Music Express Magazine [2013a; 2013b]. See the accompanying video and the supplemental document for further character control examples.

multiple interacting and overlapping subjects, including frames from MuPoTS-3D [2018b] and Panoptic [2015] datasets. Single-person real-time approaches such as VNect [2017b] are unable to handle occlusions or multiple people in close proximity as shown in Figure 5 in the supplemental document, while our approach can. Even offline single-person approaches such as HMMR [2019], which utilize multi-person 2D parsing as a pre-processing step along with temporal information, fail under occlusions. Our approach is also more robust to occlusions than the multi-person approach of Mehta et al. [2018b], and unlike top-down approaches [Dabral et al. 2019; Moon et al. 2019; Rogez et al. 2019] it does not produce spurious predictions, as seen in Figure 5 in the supplemental document. For further qualitative results on a variety of scene settings, including community videos and live scene setups, please refer to Figure 9, the accompanying video, and Figures 4 and 6 in the supplemental document.

Comparison With KinectV2: The quality of our pose estimates with a single RGB camera is comparable to off-the-shelf depth-sensing-based systems such as KinectV2 (Figure 8), with our approach succeeding in certain cluttered scenarios where person identification from depth input would be ambiguous. The accompanying video contains further visual comparisons.

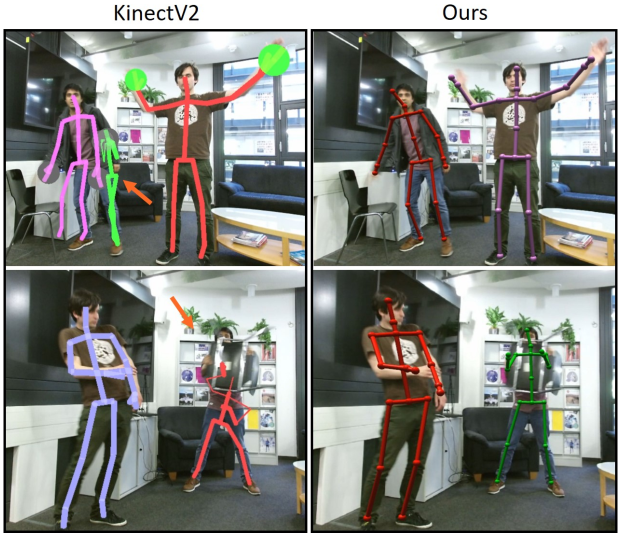

Fig. 8. The quality of our pose estimates is comparable to depth sensing based approaches such as KinectV2, and our system handles certain cases of significant inter-personal overlap and cluttered scenes better than KinectV2. In the top row, due to scene clutter, KinectV2 predicts multiple skeletons for one subject. In the bottom row, under occlusion, KinectV2 mispredicts the pose, while our approach succeeds.

Character Animation: Since we reconstruct temporally coherent joint angles and our camera-relative subject localization estimates are stable, the output of our system can readily be employed to animate virtual avatars as shown in Figure 7. The video demonstrates the stability of the localization estimates of our system and contains further examples of real-time interactive character control with a single RGB camera.

7.2 Performance on Single Person 3D Pose Datasets

Our method is capable of real-time motion capture of multi-person scenes with notable occlusions. Previous single-person approaches, irrespective of runtime, would fail on this task. For completeness, we show that our method shows competitive accuracy on single-person 3D pose estimation. In Table 4, we compare the 3D pose output after *Stage II* and *Stage III* against other single-person methods on the MPI-INF-3DHP benchmark dataset [Mehta et al. 2017a] using the commonly used 3D Percentage of Correct Keypoints (**3DPCK**, higher is better), Area under the Curve (**AUC**, higher is better) and mean 3D joint position error (**MJPE**, lower is better).

Similarly, with Stage II trained on Human3.6m (Table 3), we again see that our system compares favourably to recent approaches designed to handle single-person and multi-person scenarios. Further, this is an example of the ability of our system to adapt to different datasets by simply retraining the inexpensive *Stage II* network. Some approaches such as Sun et al. [2017; 2018], obtain state-of-the-art accuracy on Human3.6m, but critically use the camera intrinsics and the ground-truth distance of the subject from the camera to convert their (u,v) predictions to (x,y), and also use flip augmentation at test time. Therefore, their reported results are not representative of the deployed system's performance.

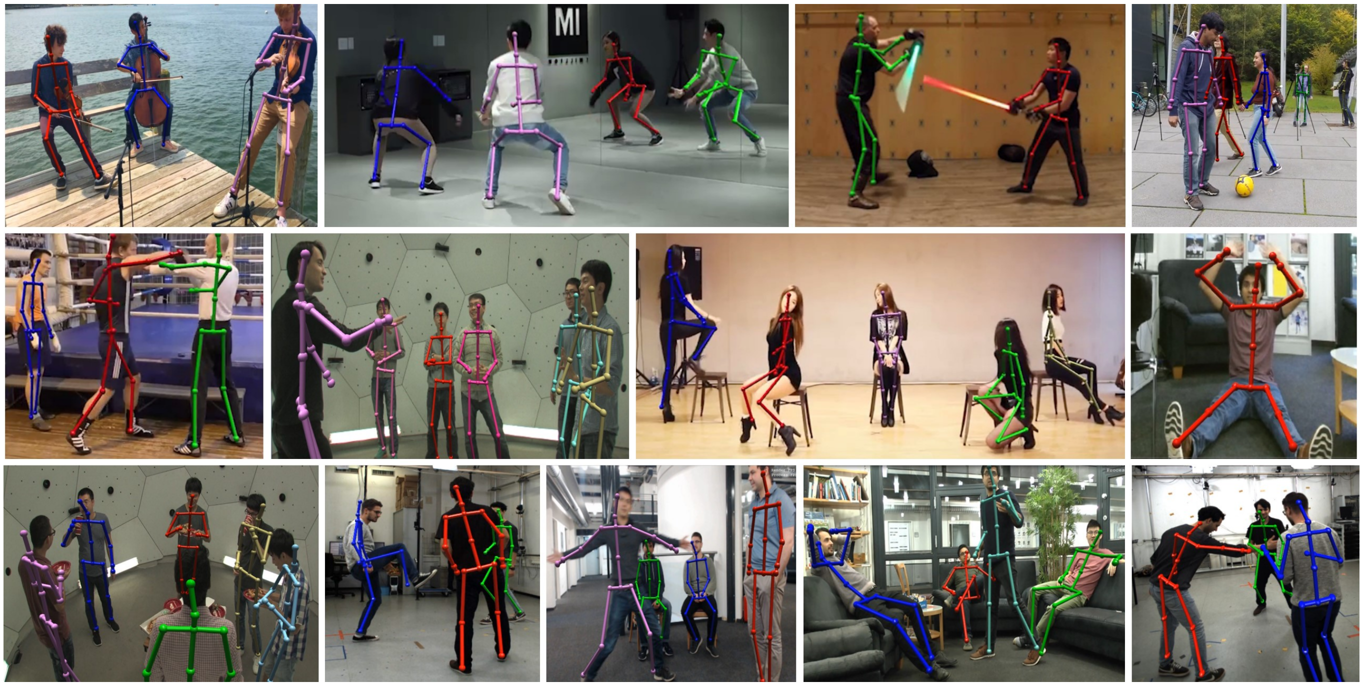

Fig. 9. Monocular 3D motion capture results from our full system (*Stage III*) on a wide variety of multi-person scenes, including Panoptic [2015] and MuPoTS-3D [2018b] datasets. Our approach handles challenging motions and poses, including interactions and cases of self-occlusion. Some images courtesy [KNG Music 2019], [1MILLION TV 2016], [Boxing School Alexei Frolov 2018], [TPLA:Terra Prime Light Armory 2016], and [Brave Entertainment 2017]. Please refer to the supplemental document and video for more results.

7.3 Performance on Multi-Person 3D Pose Datasets

We quantitatively evaluate our method's accuracy on the MuPoTS-3D monocular multi-person benchmark data set from Mehta et al. [2018b], which has ground-truth 3D pose annotations from a multi-view marker-less motion capture system. We perform two types of comparison. In Table 7(All), we compare on all annotated poses in sequences **T1-T20** of the annotated test set, including poses of humans that were not detected by our algorithm. In Table 7 (Matched), we compare only on annotated poses of humans detected by the respective algorithms.

Both tables show that our per-frame predictions (*Stage II*), as well as our temporally smooth model-fitting predictions (*Stage III*) achieve better accuracy in terms of the 3D percentage of correct keypoints metric (**3DPCK**, higher is better) to LCRNet [2017], LCRNet++ [2019] and [Mehta et al. 2018b], while being comparable to other recent methods, namely [Dabral et al. 2019]. The approach of Moon et al. [2019] exceeds the performance of our system, but it uses a prior person detection step, and passes resized person crops to the pose estimation network. As has been shown in prior work [Cao et al. 2017], such an approach results in a higher accuracy, however, it cannot run at real-time frame rates, and the run time scales almost linearly with the number of subjects in the scene. In contrast, our approach runs in real-time for multi-person scenes, and only has a very mild dependence on the number of subjects in the scene.

Note that we apply no test-time augmentation or ensembling to our system, making the reported performance on various benchmarks accurately reflect the real per-frame prediction performance of our system.

In Table 7 (Matched), we compare only on annotated poses of humans detected by the respective algorithms.

Additionally, we quantitatively evaluate our method's accuracy on the 3DPW [von Marcard et al. 2018] monocular multi-person dataset, which comprises of videos recorded in in-the-wild settings with a phone camera. The 3D pose annotations for the dataset are generated from a combination of inertial-sensing measurements and video based approaches. We use the 24 sequences in the 'test' split for evaluating. Our method does not use any data from 3DPW for training. Unlike prior and concurrent work [Kanazawa et al. 2018, 2019; Kocabas et al. 2020] which use bounding-box tracks of each subject, our approach handles all the subjects in the complete input frame jointly, and also matches person identities from frame to frame, all while running in real time. We report the mean 3D joint position error (MPJPE) in mm in Table 6. Our per-frame predictions (*Stage II*) are comparable or better than most recent approaches, while being real-time and using no bounding box information. The person tracking heuristics for *Stage III* are designed for stationary or slowly moving cameras, and the assumptions break under fast moving cameras in dense crowds where multiple subjects can have similar clothing and 2D poses. This leads to a noticeable decrease in accuracy with *Stage III*, which can be improved by incorporating a better identity tracking approach, that is less prone to failure in dense crowds. See the accompanying video for further results.

Table 3. Results of Stage II predictions on Human3.6m, evaluated on all camera views of Subject 9 and 11 without alignment to GT. The Stage II network is trained with only Human3.6m. The top part has single person 3D pose methods, while the bottom part shows methods designed for multi-person pose estimation. Mean Per Joint Position Error (MPJPE) in millimeters is the metric used (lower is better). Note that our reported results do *not use any test time augmentation*. Nor do we exploit ground truth 3D pose information in other ill-informed ways, such as rigid alignment to the ground truth or the use of ground truth depth from the camera to glean the image coordinate to 3D space transformation.

Method	MPJPE (mm)
[Katircioglu et al. 2018]	67.3
[Pavlakos et al. 2017]	67.1
[Zhou et al. 2017]	64.9
[Martinez et al. 2017]	62.9
Hossain & Little [2018]	58.3
[Mehta et al. 2018b]	69.9
LCRNet++[2019] (Res50)	63.5
LCRNet+[2019] (VGG16)	61.2
[Moon et al. 2019]	54.4
Ours (*Stage II*)	63.6

Table 4. Comparison on the single person MPI-INF-3DHP dataset. Top part are methods designed and trained for single-person capture. Bottom part are multi-person methods *trained for multi-person capture* but evaluated on single-person capture. Metrics used are: 3D percentage of correct keypoints (3DPCK, higher is better), area under the curve (AUC, higher is better) and mean 3D joint position error (MJPE, lower is better). * Indicates that *no test time augmentation is employed*. †Indicates that *no ground-truth bounding box information is used* and the complete image frame is processed.

Method	**3DPCK**	**AUC**	**MJPE**
[Mehta et al. 2017b]	78.1	42.0	119.2
[Mehta et al. 2017b]*	75.0	39.2	132.8
[Nibali et al. 2019]	87.6	48.8	87.6
[Yang et al. 2018]	69.0	32.0	-
[Zhou et al. 2017]	69.2	32.5	-
[Pavlakos et al. 2018a]	71.9	35.3	-
[Dabral et al. 2018]	72.3	34.8	116.3
[Kanazawa et al. 2018]	72.9	36.5	124.2
[Mehta et al. 2018b]	76.2	38.3	120.5
[Mehta et al. 2018b]	74.1	36.7	125.1
[Mehta et al. 2018b]*	72.1	35.1	130.3
Ours (*Stage II*)*†	82.8	45.3	98.4
Ours (*Stage III*)*†	77.8	38.9	115.0

7.4 Core Network Architecture Evaluation

We evaluate the performance of the proposed network architecture against ResNet baselines on 2D pose estimation and 3D pose estimation tasks. For ResNet, we keep the network until the first residual module in level-5 and remove striding from level-5. For 2D pose estimation, we evaluate on a held-out 1000 frame subset of the MS-COCO validation set, and report the Average Precision (AP) and Recall (AR), as well as inference time on different hardware in Table 2. The performance of SelecSLS Net (47.0 AP, 51.8 AR) is better than ResNet-50 (46.6 AP, 51.4 AR), while being 1.3 − 1.4× faster on GPUs. The inference speed improvement increases to 1.8× when running on a CPU, as shown in the supplemental document.

Table 5. Comparison of *Stage II* prediction limb joint 3D pose accuracy on MPI-INF-3DHP (Single Person) for different core network choices with our channel-sparse supervision of 3D pose branch of *Stage I*, as well as a comparison to channel-dense supervision. Metrics used are 3DPCK and AUC (higher is better).

	3DPCK				**Total**	
	Elbow	**Wrist**	**Knee**	**Ankle**	**3DPCK**	**AUC**
ResNet-34	79.6	61.2	83.0	52.7	79.3	41.8
ResNet-50	**82.4**	61.8	87.1	58.9	82.0	44.1
SelecSLS	81.2	**62.0**	**87.6**	**63.3**	**82.8**	**45.3**
Channel-Dense $\{l_{j,k}\}_{j=1}^{J}$ Supervision						
SelecSLS	79.0	60.2	82.5	59.0	80.1	43.3

Table 6. Evaluation of 3D joint position error on the 'test' split of 3DPW. The error is reported as the Mean Per-Joint Position Error (MPJPE) in mm for the following 14 joints: head, neck, shoulders, elbows, wrists, hips, knees, ankles. PA-MPJPE indicates the error after procrustes alignment. Lower is better. Note that unlike prior work [Kanazawa et al. 2018, 2019; Kocabas et al. 2020] which uses bounding-box crops around the subject and the supplied annotations to establish temporal association, our approach handles all the subjects in the complete input frame jointly, and also matches person identities from frame to frame, all while running in real time. The person identity tracking heuristics are designed for a stationary or slow moving camera, but the faster moving camera in 3DPW as well as the dense crowds cause identity tracking failures, which results in the markedly worse performance of *Stage III* compared to *Stage II*.

Method	MPJPE	PA-MPJPE
[Sun et al. 2019b]	-	69.5
[Arnab et al. 2019]	-	72.2
[Kolotouros et al. 2019a]	-	70.2
[Kanazawa et al. 2018]	130.0	76.7
[Kolotouros et al. 2019b]	96.9	59.2
[Kanazawa et al. 2019]	116.5	72.6
[Kocabas et al. 2020]	93.5	56.5
Ours(*Stage II*)	103.3	60.7
Ours(*Stage III*)	134.2	80.3

We evaluate our network trained for multi-person pose estimation (after Stage II) on MPI-INF-3DHP [Mehta et al. 2017a] single person 3D pose benchmark, comparing the performance of different core network architectures. In Table 5, we see that using our *SelecSLS* core architecture we overall perform significantly better than ResNet-34 and slightly better than ResNet-50, with a higher 3DPCK and AUC and a lower MPJPE error. *SelecSLS* particularly results in significantly better performance for lower body joints (Knee, Ankle) than the ResNet baselines.

Similarly, our *SelectSLS* network architecture outperforms ResNet-50 and ResNet-34 on the multi person 3D pose benchmark MuPoTS-3D, as shown in Table 8.

Table 7. Comparison of our per-frame estimates (Stage II) on the MuPoTS-3D benchmark data set [Mehta et al. 2018b]. The metric used is 3D percentage of correct keypoints (**3DPCK**), so higher is better. The data set contains 20 test scenes **TS1-TS20**. We evaluate once on all annotated poses (top row - **All**), and once only on the annotated poses detected by the respective algorithm (bottom row - **Matched**). Our approach achieves comparable accuracy to the previous monocular multi-person 3D methods (SingleShot [Mehta et al. 2018b], LCRNet [Rogez et al. 2017], MP3D [Dabral et al. 2019], LCRNet++ [Rogez et al. 2019]) while having a drastically faster runtime. The accuracy of our system lags behind that of Root+PoseNet [Moon et al. 2019] which uses a prior person detection step, runs offline, has its per-frame inference time scale linearly with the number of subjects in the scene. * Indicates **no** test time augmentation is used.

All	**TS1**	**TS2**	**TS3**	**TS4**	**TS5**	**TS6**	**TS7**	**TS8**	**TS9**	**TS10**	**TS11**	**TS12**	**TS13**	**TS14**	**TS15**	**TS16**	**TS17**	**TS18**	**TS19**	**TS20**	**Total**
LCRNet*	67.7	49.8	53.4	59.1	67.5	22.8	43.7	49.9	31.1	78.1	33.4	33.5	51.6	49.3	56.2	66.5	65.2	62.9	66.1	59.1	53.8
Single Shot	81.0	59.9	64.4	62.8	68.0	30.3	65.0	59.2	64.1	83.9	67.2	68.3	60.6	56.5	69.9	79.4	79.6	66.1	64.3	63.5	65.0
MP3D*	85.1	67.9	73.5	76.2	74.9	52.5	65.7	63.6	56.3	77.8	76.4	70.1	65.3	51.7	69.5	87.0	82.1	80.3	78.5	70.7	71.3
Root+PoseNet*	**94.4**	**77.5**	**79.0**	**81.9**	**85.3**	**72.8**	**81.9**	**75.7**	**90.2**	**90.4**	**79.2**	**79.9**	**75.1**	**72.7**	**81.1**	**89.9**	**89.6**	**81.8**	**81.7**	**76.2**	**81.8**
LCRNet++ (Res50)*	87.3	61.9	67.9	74.6	78.8	48.9	58.3	59.7	78.1	89.5	69.2	73.8	66.2	56.0	74.1	82.1	78.1	72.6	73.1	61.0	68.9
Ours (*Stage II*)*	89.7	65.4	67.8	73.3	77.4	47.8	67.4	63.1	78.1	85.1	75.6	73.1	65.0	59.2	74.1	84.6	87.8	73.0	78.1	71.2	72.1
Ours (*Stage III*)*	86.3	63.5	66.1	71.1	69.7	48.4	68.4	62.9	76.4	85.4	72.7	75.1	61.9	62.9	70.3	84.4	84.6	72.2	70.4	69.4	70.4

Matched	**TS1**	**TS2**	**TS3**	**TS4**	**TS5**	**TS6**	**TS7**	**TS8**	**TS9**	**TS10**	**TS11**	**TS12**	**TS13**	**TS14**	**TS15**	**TS16**	**TS17**	**TS18**	**TS19**	**TS20**	**Total**
LCRNet*	69.1	67.3	54.6	61.7	74.5	25.2	48.4	63.3	69	78.1	53.8	52.2	60.5	60.9	59.1	70.5	76	70	77.1	81.4	62.4
Single Shot	81.0	64.3	64.6	63.7	73.8	30.3	65.1	60.7	64.1	83.9	71.5	69.6	69	69.6	71.1	82.9	79.6	72.2	76.2	85.9	69.8
MP3D*	85.8	73.6	61.1	55.7	77.9	53.3	75.1	65.5	54.2	81.3	**82.2**	71.0	70.1	67.7	69.9	90.5	85.7	**86.3**	85.0	91.4	74.2
Root+PoseNet*	**94.4**	**78.6**	**79.0**	**82.1**	**86.6**	**72.8**	**81.9**	**75.8**	**90.2**	**90.4**	79.4	**79.9**	**75.3**	81.0	**81.0**	**90.7**	**89.6**	83.1	81.7	77.2	**82.5**
LCRNet++ (Res50)*	88.0	73.3	67.9	74.6	81.8	50.1	60.6	60.8	78.2	89.5	70.8	74.4	72.8	64.5	74.2	84.9	85.2	78.4	75.8	74.4	-
Ours (*Stage II*)*	89.7	78.6	68.4	74.3	83.7	47.9	67.4	75.4	78.1	85.1	78.7	73.8	73.9	77.9	74.8	87.1	88.3	79.5	**88.3**	**97.5**	78.0
Ours (*Stage III*)*	86.3	76.2	66.7	72.1	75.3	48.5	68.4	75.1	76.4	85.4	75.7	75.8	70.3	**82.9**	70.9	86.9	85.1	78.7	79.5	95.0	76.2

Table 8. Evaluations on the multi-person 3D pose benchmark MuPoTS-3D [2018b] of *Stage II* predictions for different *Stage I* core network choices with channel-sparse supervision of 3D pose branch of *Stage I*, as well as a comparison to channel-dense supervision. We evaluate on all annotated subjects using the 3D percentage of correct keypoints (**3DPCK**) metric, and also show the 3DPCK only for predictions that were matched to an annotation.

	3DPCK		**% Subjects**
	All	**Matched**	**Matched**
ResNet-34	69.3	75.2	92.1
ResNet-50	71.8	77.2	**93.0**
SelecSLS	**72.1**	**78.0**	92.5
Channel-Dense $\{l_{j,k}\}_{j=1}^{J}$ Supervision			
SelecSLS	70.2	75.7	92.7

Table 9. Comparison of limb joint 3D pose accuracy on MuPoTS-3D (Multi Person) for predictions from *Stage II* and *Stage III* of our proposed design. The metric used is 3D Percentage of Correct Keypoints (3DPCK), evaluated with a threshold of 150mm.

						All Joints		
	Shoul	**Elbow**	**Wrist**	**Knee**	**Ankle**	**Total**	**Vis.**	**Occ.**
Stage II	**81.4**	65.8	53.2	71.0	47.3	**72.1**	**76.0**	**58.8**
Stage III	73.8	**67.2**	**54.0**	**75.8**	**54.3**	70.4	75.0	55.2

With *Stage II* trained on the single-person pose estimation task on Human3.6m, we again see that our proposed faster core network architecture outperforms ResNet baselines. The use of *SelecSLS* results in a mean per joint position error of 63.6mm, compared to 64.8mm using ResNet-50 and 67.6mm using ResNet-34.

7.5 Channel-Sparse 3D Pose Encoding Evaluation

As discussed at length in Section 4.1.2, different choices for supervision of $\{l_{j,k}\}_{j=1}^{J}$ have different implications. Here we show that our channel-sparse supervision of the encoding, such that only the local kinematic context is accounted for, performs better than the naïve channel-dense supervision.

The 2D pose accuracy of the *Stage I* network with channel-dense supervision of 3D branch is comparable to that with channel-sparse supervision, as shown in Table 2. However, our proposed encoding performs much better across single person and multi-person 3D pose benchmarks.

Table 5 shows that channel-sparse encoding significantly outperforms channel-dense encoding, yielding an overall 3DPCK of 82.8 compared to 80.1 3DPCK for the latter. As shown in Table 4 in the supplemental document, the difference particularly emerges for difficult pose classes such as sitting on a chair or on the floor, where our channel-sparse supervision shows substantial gains. Also refer to the supplemental document for additional ablative analysis of input to *Stage II*.

7.6 Evaluation of Skeleton Fitting (Stage III)

Skeleton fitting to reconcile 2D and 3D poses across time results in smooth joint angle estimates which can be used to drive virtual characters. The accompanying video shows that the resulting multi-person pose estimates from our real-time system are similar or better in quality than the recently published offline single-person approach of [Kanazawa et al. 2019], while also succeeding under significant occlusions.

On the multi-person benchmark MuPoTS-3D in Table 7, the overall performance of *Stage III* appears to lag behind that of *Stage II*. However, the efficacy of *Stage III* is revealed through joint-wise breakdown of performance on MuPoTS-3D in Table 9. *Stage III* predictions show a significant improvement for limb joints such as elbows, wrists, knees, and ankles over *Stage II*. The decrease in accuracy for some other joints, such as the head, is partly due to lack of 2D constraints imposed for those joints in our system. The overall breakdown of accuracy by joint visibility shows that *Stage III* improves the accuracy for visible joints, i.e. where 2D constraints were available, while slightly decreasing accuracy for occluded joints.

On the single-person benchmark MPI-INF-3DHP (Table 4), the performance of *Stage III* is comparable to other methods specifically designed for single-person scenarios, however it loses accuracy compared to *Stage II* predictions for similar reasons as in the multi-person case. For complex poses, not all 2D keypoints may be detected. This does not pose a problem for *Stage II* predictions due to redundancies in the pose encoding, however, analogous to the case of occluded joints discussed before, missing 2D keypoints can cause the *Stage III* accuracy to worsen.

However, despite a minor decrease in quantitative accuracy due to *Stage III*, it produces a marked improvement in the quality and temporal stability of the pose estimates, and increases the accuracy of the end effectors. The resulting, temporally smooth joint angle estimates can be used in interactive graphics applications. Refer to the accompanying video for qualitative results, and virtual character control examples.

8 DISCUSSION AND FUTURE WORK

Our approach is the first to perform real-time 3D motion capture of challenging multi-person scenes with one color camera. Nonetheless, it has certain limitations that will be addressed in future work.

As with other monocular approaches, the accuracy of our method is not comparable yet to the accuracy of multi-view capture algorithms. Failure cases in our system can arise from each of the constituent stages. The 3D pose estimates can be incorrect if the underlying 2D pose estimates or part associations are incorrect. Also, since we require the neck to be visible for a successful detection of a person, scenarios where the neck is occluded result in the person not being detected despite being mostly visible. See Figure 3 in the supplemental document for a visual representation of the limitations.

Our algorithm successfully captures the pose of occluded subjects even under difficult inter-person occlusions that are generally hard for monocular methods. However, the approach still falls short of reliably capturing extremely close interactions, like hugging. Incorporation of physics-based motion constraints could further improve pose stability in such cases, may add further temporal stability, and may allow capturing of fine-grained interactions of persons and objects.

In some cases individual poses have higher errors for a few frames, e.g. after strong occlusions (see accompanying video for example). However, our method manages to recover from this. The kinematic fitting stage may suffer from inaccuracies under cases of significant inter-personal or self occlusion, making the camera relative localization less stable in those scenarios. Still, reconstruction accuracy and stability is appropriate for many real-time applications.

Our algorithm is fast, but the relatively simple identity tracker may swap identities of people when tracking through extended person occlusions, drastic appearance changes, and similar clothing appearance. More sophisticated space-time tracking would be needed to remedy this. As with all learning-based pose estimation approaches, pose estimation accuracy worsens on poses very dissimilar from the training poses. To approach this, we plan to extend our algorithm in future such that it can be refined in an unsupervised or semi-supervised way on unlabeled multi-person videos.

Our *SelecSLS Net* leads to a drastic performance boost. There are other strategies that could be explored to further boost the speed of our network and convolutional architectures in general, or target it to specific hardware, such as using depthwise 3×3 convolutions or factorized 3×3 convolutions [Romera et al. 2018; Szegedy et al. 2017] or binarized operations [Bulat and Tzimiropoulos 2017], all of which our proposed design can support. Its faster inference speed and significantly smaller memory footprint than ResNet-50 without compromising accuracy makes it an attractive candidate to replace ResNet core networks for a broad range of tasks beyond body pose estimation. It can support much larger training and inference batch sizes, or can be trained on lower-end GPUs with similar batch sizes as ResNet-50 on higher-end GPUs.

9 CONCLUSION

We present the first real-time approach for multi-person 3D motion capture using a single RGB camera. It operates in generic scenes and is robust to occlusions both by other people and objects. It provides joint angle estimates and localizes subjects relative to the camera. To this end, we jointly designed pose representations, network architectures, and a model-based pose fitting solution, to enable real-time performance. One of the key components of our system is a new CNN architecture that uses selective long and short range skip connections to improve the information flow and have a significantly smaller memory footprint, allowing for a drastically faster network without compromising accuracy. The proposed system runs on consumer hardware at more than 30 fps while achieving state-of-the-art accuracy. We demonstrate these advances on a range of challenging real-world scenes.

ACKNOWLEDGMENTS

This work was funded by the ERC Consolidator Grant 4DRepLy (770784). We thank Gereon Fox, Abhimitra Meka, Ikhsanul Habibie, Vikram Singh, Marc Habermann, Neng Qian, Jiayi Wang, Vladislav Golyanik, Edgar Tretschk, Tarun Yenamandra, Lingjie Liu, and Ayush Tewari for helping with data capture and the preparation of video materials.

REFERENCES

1MILLION TV. 2016. GDFR - Flo Rida / 1MILLION Dance TUTORIAL (1/2). https://www.youtube.com/watch?v=9HkVnFpmXAw.

Adobe. 2020. Mixamo. https://www.mixamo.com/.

Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to Reconstruct People in Clothing from a Single RGB Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018a. Detailed Human Avatars from Monocular Video. In *3DV*.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018b. Video Based Reconstruction of 3D People Models. In *CVPR*.

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *CVPR*.

Anurag Arnab, Carl Doersch, and Andrew Zisserman. 2019. Exploiting temporal context for 3D human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3395–3404.

Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. 2019. Multi-Garment Net: Learning to Dress 3D People from Images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *ECCV*.

Boxing School Alexei Frolov. 2018. Boxing: bouncing - slip - overhand punch. https://www.youtube.com/watch?v=dbuz9Q05bsM.

Brave Entertainment. 2017. (Rollin') Dance Practice Video. https://www.youtube.com/watch?v=ZhuDSdmby8k.

Adrian Bulat and Georgios Tzimiropoulos. 2017. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *International Conference on Computer Vision*.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 &#8364; Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems *(CHI '12)*. ACM, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

Jinxiang Chai and Jessica K Hodgins. 2005. Performance animation from low-dimensional control signals. *TOG* 24, 3 (2005), 686–696.

Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2016. Synthesizing Training Images for Boosting Human 3D Pose Estimation. In *3DV*.

Rishabh Dabral, Nitesh B Gundavarapu, Rahul Mitra, Abhishek Sharma, Ganesh Ramakrishnan, and Arjun Jain. 2019. Multi-Person 3D Human Pose Estimation from Monocular Images. In *2019 International Conference on 3D Vision (3DV)*. IEEE, 405–414.

Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. 2018. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 668–683.

A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. 2016. MARCOnI - ConvNet-based MARker-less Motion Capture in Outdoor and Indoor Scenes. *PAMI* (2016).

Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. 2018. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635* (2018).

Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. 2014. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*. 3582–3589.

Peng Guan, A. Weiss, A. O. BÃčlan, and M. J. Black. 2009. Estimating human shape and pose from a single image. In *CVPR*. 1381–1388. https://doi.org/10.1109/ICCV.2009.5459300

Riza Alp Guler and Iasonas Kokkinos. 2019. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10884–10894.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. DensePose: Dense Human Pose Estimation in the Wild. In *CVPR*. 7297–7306. https://doi.org/10.1109/CVPR.2018.00762

Semih Gunel, Helge Rhodin, and Pascal Fua. 2019. What Face and Body Shapes Can Tell Us About Height. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.

Marc Habermann, Weipeng Xu, , Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2019. LiveCap: Real-time Human Performance Capture from Monocular Video. *ACM Transactions on Graphics, (Proc. SIGGRAPH)* (jul 2019).

Lei Tan Lin Gui Bart Nabbe Iain Matthews Takeo Kanade Shohei Nobuhara Hanbyul Joo, Hao Liu and Yaser Sheikh. 2015. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *ICCV*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*. 1314–1324.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017b. Densely connected convolutional networks.. In *CVPR*.

Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. 2017a. Towards accurate marker-less human shape and pose estimation over time. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 421–430.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv:1602.07360* (2016).

Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, Bernt Schiele, and Saarland Informatics Campus. 2017. ArtTrack: Articulated multi-person tracking in the wild. In *CVPR*.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI* 36, 7 (2014), 1325–1339.

Umar Iqbal and Juergen Gall. 2016. Multi-person pose estimation with local joint-to-person associations. In *ECCV Workshops*. Springer, 627–642.

Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. 2010. MovieReshape: Tracking and Reshaping of Humans in Videos. *TOG* 29, 6, Article 148 (Dec. 2010), 10 pages. https://doi.org/10.1145/1882261.1866174

Sam Johnson and Mark Everingham. 2010. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *BMVC*. doi:10.5244/C.24.12.

Sam Johnson and Mark Everingham. 2011. Learning Effective Human Pose Estimation from Inaccurate Annotation. In *CVPR*.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2015. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *ICCV*. 3334–3342.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *CVPR*.

Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3D Human Dynamics from Video. In *Computer Vision and Pattern Recognition (CVPR)*.

Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2018. Learning latent representations of 3d human pose with deep neural networks. *International Journal of Computer Vision* 126, 12 (2018), 1326–1341.

KNG Music. 2019. SEÃŚORITA - Shawn Mendes Camila Cabello | Violin, Cello, and Viola Cover [KNG Music]. https://www.youtube.com/watch?v=_xCKmEhKQl4.

Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. 2019b. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *Proceedings of the IEEE International Conference on Computer Vision*. 2252–2261.

Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. 2019a. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4501–4510.

Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. 2017. Unite the People: Closing the Loop Between 3D and 2D Human Representations. In *CVPR*.

Sijin Li and Antoni B Chan. 2014. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*.

Sijin Li, Weichen Zhang, and Antoni B Chan. 2015. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *ICCV*. 2848–2856.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*. Springer, 740–755.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *TOG* 34, 6 (2015), 248:1–248:16.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 116–131.

Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *ICCV*.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017a. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In *3DV*. IEEE. https://doi.org/10.1109/3dv.2017.00064

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. 2018b. Single-Shot Multi-Person 3D Pose Estimation From Monocular RGB. In *3DV*. IEEE. http://gvv.mpi-inf.mpg.de/projects/SingleShotMultiPerson

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017b. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. In *TOG*, Vol. 36. 14. https://doi.org/10.1145/3072959.3073596

Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. 2018a. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In *ECCV*.

Alberto Menache. 2010. *Understanding Motion Capture for Computer Animation, Second Edition* (2nd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. 2019. Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. In *The IEEE Conference on International Conference on Computer Vision (ICCV)*.

Music Express Magazine. 2013a. Arirang. https://www.youtube.com/watch?v=kX6xMYlEwLA.

Music Express Magazine. 2013b. Down at the Twist Shout. https://www.youtube.com/watch?v=lv-h4WNnw0g.

Alejandro Newell and Jia Deng. 2017. Associative Embedding: End-to-End Learning for Joint Detection and Grouping. In *NeurIPS*.

Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 2019. 3D Human Pose Estimation with 2D Marginal Heatmaps. In *WACV*.

Mohamed Omran, Christop Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. 2018. Neural Body Fitting: Unifying Deep Learning and Model Based Human Pose and Shape Estimation. In *3DV*.

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. 2017. Towards Accurate Multi-person Pose Estimation in the Wild. In *CVPR*.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. (2019).

Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. 2018a. Ordinal Depth Supervision for 3D Human Pose Estimation. In *CVPR*.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *CVPR*.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. 2018b. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *CVPR*.

Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. 2016. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *CVPR*.

Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. 2012. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*. IEEE, 3178–3185.

Gerard Pons-Moll, David J Fleet, and Bodo Rosenhahn. 2014. Posebits for monocular human pose estimation. In *CVPR*. 2337–2344.

Alin-Ionut Popa, Mihai Zanfir, and Cristian Sminchisescu. 2017. Deep Multitask Architecture for Integrated 2D and 3D Human Sensing. In *CVPR*.

Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2012. Reconstructing 3d human pose from 2d image landmarks. In *ECCV*. Springer, 573–586.

Mir Rayat Imtiaz Hossain and James J Little. 2018. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 68–84.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. 33 (2019), 4780–4789.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*. 91–99.

Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016a. EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras. *TOG (Proc. SIGGRAPH Asia)* (2016).

Helge Rhodin, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2016b. General Automatic Human Shape and Motion Capture Using Volumetric Contour Cues. In *ECCV*. 509–526. https://doi.org/10.1007/978-3-319-46454-1_31

Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2017. LCR-Net: Localization-Classification-Regression for Human Pose. In *CVPR*.

Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2019. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *PAMI* (2019).

Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. 2018. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems* 19, 1 (2018), 263–272.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*. IEEE, 4510–4520.

Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A Kakadiaris. 2016. 3D Human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding* 152 (2016), 1–20.

Leonid Sigal, Alexandru O Balan, and Michael J Black. 2010. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV* 87, 1-2 (2010), 4–27.

Jonathan Starck and Adrian Hilton. 2007. Surface capture for performance-based animation. *IEEE computer graphics and applications* 27, 3 (2007).

Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. 2011. Fast articulated motion tracking using a sums of Gaussians body model. In *ICCV*. 951–958.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019a. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*.

Min Sun and Silvio Savarese. 2011. Articulated part-based model for joint object detection and pose estimation. In *ICCV*. IEEE, 723–730.

Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. 2017. Compositional human pose regression. 2, 3 (2017), 7.

Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. 2018. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 529–545.

Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, Yili Fu, and Tao Mei. 2019b. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *Proceedings of the IEEE International Conference on Computer Vision*. 5349–5358.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning.. In *AAAI*, Vol. 4. 12.

Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2019).

Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2016. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *BMVC*.

Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. 2017. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *ICCV*.

Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. 2019. xR-EgoPose: Egocentric 3D Human Pose from an HMD Camera. In *Proceedings of the IEEE International Conference on Computer Vision*. 7728–7738.

TPLA:Terra Prime Light Armory. 2016. Commited Sparring. https://www.youtube.com/watch?v=xmFVfUKr1MQ.

Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *BMVC*. 1–13.

Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. 2017. Self-supervised Learning of Motion Capture. In *NeurIPS*. 5242–5252.

Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In *ECCV*.

Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. 2016. Human Pose Estimation from Video and IMUs. *PAMI* 38, 8 (Jan. 2016), 1533–1547.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. 2019. Deep High-Resolution Representation Learning for Visual Recognition. *TPAMI* (2019).

Xiaolin Wei and Jinxiang Chai. 2010. VideoMocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences. *TOG* 29, 4, Article 42 (July 2010), 10 pages. https://doi.org/10.1145/1778765.1778779

Shihong Xia, Lin Gao, Yu-Kun Lai, Mingzhe Yuan, and Jinxiang Chai. 2017. A Survey on Human Performance Capture and Animation. *J. Comput. Sci. Technol.* 32, 3 (2017), 536–554.

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. 2019. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10965–10974.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *CVPR*. IEEE, 5987–5995.

Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. 2019. Mo 2 Cap 2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2093–2101.

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. Monoperfcap: Human performance capture from monocular video. *TOG* 37, 2 (2018), 27.

Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 2018. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1.

Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. 2018a. Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes–The Importance of Multiple Scene Constraints. In *CVPR*. 2148–2157.

Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. 2018b. Deep Network for the Integrated 3D Sensing of Multiple People in Natural Images. In *NeurIPS*.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6848–6856.

Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. 2017. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach. In *CVPR*. 398–407.