Towards Holistic Real-time Human 3D Pose Estimation using MocapNETs

Ammar Qammaz ammarkov@ics.forth.gr

Antonis A. Argyros argyros@ics.forth.gr Computer Science Department, University of Crete, Heraklion, Crete, Greece Institute of Computer Science, FORTH, N. Plastira 100, Vassilika Vouton, GR70013, Heraklion, Crete, Greece

Abstract

In this work, we extend a method originally devised for 3D body pose estimation to tackle the 3D hand pose estimation task. Due to its compositionality and compact Bio Vision Hierarchy (BVH) output, the resulting method can be combined with the original 3D body pose estimation method. This is achieved based on a novel neural network architecture combining key design characteristics of DenseNets, ResNets and MocapNETs trainable to accommodate both bodies and hands. The resulting method is assessed quantitatively in well-established hand and body pose estimation datasets. The obtained results show that the proposed enhancements result in competitive performance for hands, as well as on accuracy and performance benefits for the original body estimation task. Moreover, we show qualitatively that due to its real-time performance and easy deployment using off-the-shelf webcam equipped PCs, the proposed solution can become a valuable perceptual building block supporting a variety of applications.

1 Introduction

After decades of research, during 2020 we witnessed a boom in commercial hand tracking with affordable tether-less VR headsets that included ego-centric 3D hand pose estimation capabilities [1] purchased by millions new users after the debut of Oculus Quest 2. However, full-body pose estimation is still a lacking feature as well as hand estimation when hands are outside the relatively narrow view of headset cameras forcing popular new VR platforms like VRChat [1] to adopt bulky and expensive active tracking systems [1].

In this paper, we present a method that estimates the articulated 3D pose of both hands and bodies from monocular RGB sources in real-time. A formulation that generalizes over different body parts is an important goal first described by the pioneering 1994 work of Rehg and Kanade [23]. For our baseline we chose the MocapNET body tracker [23, 23] due to its extensible architecture that had sufficient representational capabilities and performance head-room to allow such a goal to be achieved in real-time. After introducing several novel aspects, we tailor it to the 3D hand pose estimation task. Methodological improvements over the baseline are multiple. Compared to the quadruple memory and training requirements of [23] we achieve substantial savings by overcoming the need for orientation classification [23, 23]. Our enhanced normalized signed rotation matrix (eNSRM) better encodes

© 2021. The copyright of this document resides with its authors.

It may be distributed unchanged freely in print or electronic forms.



Figure 1: Method outline, series and concurrency of steps from an RGB image to a 3D body + hands BVH pose. Both hands are regressed via the same ensemble due to their symmetry.

pose structure while retaining the parameter count of the baseline NSRM [2]. Moreover, we perform semi-supervised training of our ensembles using the hand kinematic limits in contrast to the fully-supervised training of [2, 2] and handle both hands with the same network leveraging 2D symmetries. To the best of our knowledge our 3D hand estimator is the first to directly output Bio Vision Hierarchies (BVH) [5] and is thus compatible with 3D animation software. Our mean hand pose estimation error is 9.9mm on the STB [2] dataset.

Back-porting our improvements to the original body estimation problem yields enhanced results. We achieve a 9% accuracy boost compared to [[1]] in Human 3.6M (H36M) [[1]] with a quarter of runtime memory and training time requirements. Our average CPU body tracking execution time is 106Hz on an i7-4790 CPU an improvement from the 70Hz of [[1]].

Our hand pose estimator is then integrated to the body estimator. The encoder ensemble we propose offers a novel dense architecture deepened using residual layers and generalizes to accommodate both hands and body. The resulting ensemble output is not raw 3D points but relative 3D angles across the kinematic hierarchy of all joints of each hand and, using the full ensemble, the rest of the body. The complete proposed ensemble (Figure 1) handles hands and body in CPU only execution with 2D to 3D regression frame-rates of 60Hz.

Our formulation offers unique advantages compared to other methods. Its modularity permits execution of parts or all of it depending on the visibility of the subject. Occluded ensemble sub-hierarchies can be organically culled from execution without interfering with estimation for visible parts and scale changes can be handled in the same manner. When users are far from the camera and sensor resolution does not allow clear, high-confidence 2D joint estimations, execution can be gracefully skipped without causing erroneous 3D output.

To showcase the flexibility of our architecture we attempt to accommodate various challenging pose estimation scenarios. A particularly compelling application is VR while others include affordable 3D motion capture, sign language recognition, sports and online videos.

2 Related Work

Decades of research in neural networks [53] yielded methods that robustly dealt with the problem of classification in RGB images [53]. Building on these foundations, categorization classifiers gradually specialized to human bodies with works like DeepPose [52] providing joint regressors that localized and pinpointed body joints at specific parts of RGB images. Methods like OpenPose [12], [23] expanded the scope of the 2D networks to accommodate bodies, hands and faces. Recent developments [23, 13] led to Total 3D Capture techniques [100] that manage to tackle the combined RGB-to-3D problem. Until recently, 3D pose estimation research focused on specific body parts, so we will attempt to address the literature likewise. A recent survey of state-of-the-art deep-learning methods for monocular body pose estimation from RGB is provided by Chen et al. [23]. For hands, the survey of Li

et al. [D] offers a comprehensive collection of contemporary hand pose estimation methods. **3D hand pose estimation**: Notable field milestones include [77] that utilized GPGPU acceleration and the Kinect RGB-D sensor to tackle 3D hand tracking in real-time using Particle Swarm Optimization (PSO) [1]. RGB-D methods dominated the field and algorithmic strategies where mainly divided to model-based generative approaches and data-driven discriminative ones. Our method uses both concepts thus falling in the "hybrid" category. Noteable generative optimization algorithms are Particle Swarm Optimization (PSO) [1], Iterative Closest Point (ICP) [1], Levenberg-Marquardt [1] and Gauss-Newton [1] among others, while discriminative algorithms include Support Vector Machines [11] and Random Forests [12]. Depth data allowed many 3D hand tracking methods to utilize nearest object or blob segmentation $[\mathbf{III}, \mathbf{\Sigma}]$, $\mathbf{\Sigma}$ while others utilized RGB skin detection $[\mathbf{\Sigma}], \mathbf{\Sigma}$ or markers [12] to a similar end. The advent of NNs caused a transition to RGB that for hands started taking place in 2014 after the influential work of Tompson et al. [22] that presented a convolutional neural network (CNN) for 2D joint localization leading to methods like Zimmermann et al. [11] that tackled 3D hand pose estimation from RGB paying the way for recent NN methods [44, 53, 54, 94] that continuously improve the state of the art.

3D body pose estimation: Notable works include LCR-Net [72], 73], VNect [52], XNect [53] and a multitude of other neural network (NN) approaches to tackle the problem [72, 79, 71, 71], 73] that led to more recent works like MoVNect [73], MotioNet [82] and PoP-Net [53].

We use the core architecture of MocapNET [12], [2] which we adapt to a different domain to create a novel 3D hand pose estimator, improve in terms of accuracy and performance on the original body estimation task and fuse in a hand+body method that handles both. The conceptually closest combined hand+body methods that perform the same task as we, are "Monocular Total Capture" [101] and Frank Mocap Hand+Body [20], however we deal with missing subhierarchies, achieve superior computational performance and output accessibility. 3D output models employed by methods exhibit incredible variety, since most use their own internal models. Popular common choices are MANO [12] for hands and SMPL [52] for bodies. Some use combinations like [111] with SMPL + frankenstein hand model [12]. Our method uses the BVH [53] open standard with a motionbuilder [12] armature. We can thus render our results using popular open-source tools like Blender [9] and MakeHuman [51].

3 Methodology

An illustrated outline of our method is presented in Figure 1. RGB frames from a monocular camera are first converted to 2D joints using a real-time body+hand 2D joint estimator. For our experiments we selected OpenPose [29] due to its robustness. The estimated 2D joints are then encoded as enhanced Normalized Signed Rotation Matrices (eNSRMs), a descriptor analyzed in the following paragraphs, and in-turn fed to NN ensembles that directly derive a 3D pose in a BVH container. Since every encoder of our ensembles (Figure 2) is conditionally independent, we expect only sporadic regression errors on the vector of BVH results. To correct these imperfections and enable personalisation for specific hand/body types and camera setups we perform a final fine tuning step using Hierarchical Coordinate Descent (HCD) [23]. The resulting output vector can be immediately consumed by an application, stored as a BVH file or be transmitted over WiFi to a device like a VR-headset.

Enhanced Normalized Signed Rotation Matrices (eNSRMs) for bodies:

We observe that the Normalized Signed Rotation Matrices (NSRMs) [2] of the baseline method have diagonal elements that are not utilized. We proceed to fill them with distance features in an attempt to enrich the descriptor without inflating parameter count. We coin the



Figure 2: A NN encoder that regresses one d.o.f. of our problem. 319 hand 2D joint and eNSRM input values are regressed through 12 densely connected layers to a single BVH rotation angle. All layers except the final linear layer use SWISH [12] activations. Concatenating 34 such encoders with some structural constants yields a full BVH 3D hand.

"enhanced" NSRM descriptor as "eNSRM" to disambiguate it from the original NSRM.

An eNSRM associated with M joints is derived as follows. The coordinates (a_x, a_y) of each used 2D joint a are normalized to the input frame dimensions and are thus bounded in the range [0, 1]. Each joint is also associated with a visibility parameter a_v provided by thresholding the 2D joint confidence values (with a 0 marking an occluded joint and 1 a visible joint). For each pair of visible 2D joints $a = (a_x, a_y), b = (b_x, b_y)$ we can define a new point $c = (b_x, b_y - | b - a |)$ that is the point b translated vertically by the length of vector ab. Using the three points a, b, c we can encode the relation between points a and b as well as their relative rotation towards a fixed vertical axis as follows:

$$eNSRM(a,b) = \begin{cases} tan^{-1}((|\vec{ab} \times \vec{cb}|)/(\vec{ab} \cdot \vec{cb}) & a \neq b, \\ |\vec{aR}| & otherwise, \end{cases}$$
(1)

where \cdot and \times denote inner and cross products, respectively. Each resulting eNSRM(a,b)angle is invariant to 2D point cloud translation and scale. The representation encodes the relative position of joints (albeit using the rotation formed from triangle $a\hat{b}c$), as well as orientation (since bc is parallel to the Y axis of the world). In contrast to the original NSRM [derivation, diagonal matrix elements (a = b) contain Euclidean distances from the R root joint, better encoding features that provide hints for the scale of the observed points introducing some resemblance to encoded Euclidean distances of EDMs [100] and NSDMs [102]. Enhanced Normalized Signed Rotation Matrices (eNSRMs) for hands: Our hand model follows the MakeHuman [1] skeleton definitions and each of its five fingers consists of four joints. We use the final three joints of each finger plus the wrist location to build our eNSRM. This joint selection yields a total of M = 16 2D input points $J_{2D} = \{p_1, ..., p_{16}\}$. Out of those, we treat the wrist p_1 as our special root point R and the rotation angle of the wrist to middle finger proximal phalanx compared to the world Y axis as our R_r root rotation. The eNSRM matrix we described above was capable to be used as a body pose descriptor in constrained orientation scenarios like 2 way [22] or 4 way [23] orientation grouping. However, hands exhibit a much larger variety of 3D rotations in space compared to human torsos that are typically upright. In our effort to disencumber the neural networks from the harder task, we perform an affine 2D rotation of all 2D joints using the inverse root point orientation $-R_r$. This brings the wrist to middle finger vector to an upright orientation parallel to the Y world axis. We record R_r and store it to the first (previously empty) diagonal eNSRM element. Using this transformation hand poses become 2D rotation invariant except for their first element. The problem resolved with eNSRMs can be better understood with the following example. Assuming a hand pose, if we perform affine rotation of the observed 2D joints, we get NSRM matrices with large changes in all elements even if no articulation change occurred. All this variability has to be correctly handled by the NN despite essentially encoding the same pose. *eNSRMs* are decorrelated from 2D rotation while allowing the absolute 3D rotation of the hand to remain retrievable by the NN via the first element.

Overcoming the need for orientation classification: A major architectural design decision in MocapNETs [22, 23] is to partition pose space in bounded orientation classes. This permits training of ensembles that deal with subsets of the 3D pose estimation problem. As a result, NN encoders become better specialized in distinguishing between similar poses. The original MocapNET method [2] featured 2 partitions, front and back, while its improved version [12] split orientations in 4 quadrants further reducing the scope of the problem handled by each ensemble. Our initial tests followed the 4-way partitioning of [2]. However, it quickly became apparent that even with the eNSRM descriptor that mitigates 2D rotation variances, observed hands exhibited a much larger root 3D orientation variability compared to the body pose problem. Initial efforts to treat the problem in a divide-and-conquer fashion like [72], [73] led us to partition orientation space in 3D using an icosahedron solid with the goal of training an ensemble for each of its twenty faces. Despite the best of our efforts this approach overly encumbered the classifier creating a central point of failure for the pose estimation of each hand. The best orientation classifier we where able to train achieved a mediocre 82% classification accuracy on our randomized training data. Furthermore, training times and runtime memory demands increased twentyfold. Thus, we opted to forego orientation classification and instead, increase the capacity of the ensemble.

The neural network ensemble: General NN improvement strategies include deepening and increasing layer parameter counts for improved network capacity. Attempting this in a densely connected network like $[\square, \square]$ however is not straightforward. Dense network parameter counts increase exponentially with added hidden layers and their ensembles quickly become bulky, difficult to train and slow to execute. At the same time, an other adverse effect of "full layer connectivity" is over-fitting. Even applying strong regularization via dropout [\square] at a high rate of 30% we get diminishing returns after 5 dense layers.

Techniques to deepen networks typically improve information flow like Highway Networks [56]. Works like Deep Residual Learning [53] create residual learning blocks that achieve this effect via alternative data paths or similarly [52] consists of dense blocks punctuated by convolutional and pooling layers. Keeping these works in mind and influenced by the study of Bianco et al. [52] that highlights the high accuracy and low resource consumption on ResNets and Densenets, we attempt to fuse and combine their designs. After numerous experiments with variations of the original architecture [52] we successfully extend MocapNET encoders by stacking dense layers that feature residual connections to remedy vanishing gradients. We retain the beneficial high dropout [52] to deal with over-fitting and partly corrupted/missing input data in cases of occlusions. Figure 2 illustrates our proposed architecture, a hybrid of the above mentioned design elements.

Our proposed architecture also employs the SWISH [$\boxed{\mbox{C2}}$] activation function. This consistently improved training accuracy compared to the SNN [$\boxed{\mbox{C2}}$] function of [$\boxed{\mbox{C2}}$, $\boxed{\mbox{C3}}$] by 0.1° up to 1° in each recovered d.o.f. During the course of development of our method we also experimented with MISH [$\boxed{\mbox{C2}}$], however despite its superior accuracy in benchmarks like Cifar-10 [$\boxed{\mbox{C3}}$] we found it not to perform better in our problem.

Each proposed hand encoder (Figure 2) has 149,288 parameters. Although the BVH hand armature we use consists of 73 d.o.f, some joints are structural while other d.o.f. are immobile due to the human bone and tendon structure. For example, human fingers can't twist. Thus 34 trained encoder outputs for the aggregate hand ensemble that use $\approx 5.0M$ parameters (a number suitable for CPU execution) are enough to regress a 3D hand.

Using the same architecture for the upper/lower body pose ensembles, the aggregate body



Figure 3: Our method accurately tracks the variety of hand poses featured on the STB dataset [22] achieving a mean average 3D joint error of 9.9mm.

ensemble amounts to $\approx 13M$ parameters a number on par with the employed baseline [$\Box 3$]. This is the case because our eNSRM descriptor features the same total number of input elements with NSRMs and because the layers and residual connections we introduce are added to a relatively narrow part of the encoders that does not overly inflate parameter count.

Regressing more than one d.o.f. with a single encoder: Each employed encoder (Figure 2) regresses 1 d.o.f of the BVH output. This design decision may initially look counter-intuitive given that we aim at a compact real-time system. What is more, since all encoders share the same inputs, it is reasonable to assume that subgroups of initial hidden layers of the final ensemble may encode common intermediate representations of the input that could possibly be merged (similar to autoencoder [6] internal organization). To investigate potential room for improvement we attempted regressing multiple d.o.f. via a single encoder with experiments summarized in Table 4. We regress 1, 2, 3 and 5 d.o.f. using a single encoder, and measure the impact on accuracy achieved on the STB dataset without the HCD step for clear results. We observe that accuracy degrades as more outputs are added. The average 3D joint error increases by +2mm for groups of 2 d.o.f., +7mm for 3 d.o.f. and +9mm for 5 d.o.f. Accommodating 2x, 3x, or 5x times the d.o.f using the same number of weights is harder both due to the finite network capacity as well as due to training difficulties. In contrast to classification tasks where neural networks typically use categorical cross entropy loss functions, all values have the same magnitude and only one of the the outputs is active for each sample, in our 3D pose regression problem all values fluctuate in different rates and ranges at the same time and not necessarily in a correlated fashion. Since the loss function for our regression problem is MSE, calculated gradients and therefore the whole mini-batch SGD [59] process will be negatively impacted due to mismatched outputs grouped together. For example, grouping 2 outputs on each encoder, the second encoder will have to regress both a BVH z-position value in millimeters as well as a rotation value of the hand that has a resolution that is orders of magnitudes smaller. More outputs make this training problem increasingly ill-posed. Although there is theoretical interest on the topic of multiple target regression [3] along with potential benefits, the ensemble architecture with encoders that handle one d.o.f. each proves a sensible choice for our problem that allows finer training control.

Leveraging hand symmetries: We observe that left and right hands have the same 2D projections when mirrored. We leverage this total symmetry on the X axis to our advantage. Instead of dealing with each one as a separate entity we perform an X flip on the right hand normalized 2D coordinates, by converting each point (a_x, a_y) to $(1 - a_x, a_y)$. After this operation, a 2D right hand appears identical to a 2D left hand and we can proceed to execute the rest of the pipeline and regress it as such. The BVH output of the NN needs sign inversions in the X position, Y rotation and all finger articulations to be converted back to a 3D right hand output. An advantage of this approach is no bias while treating left or right hands since they are regressed with the same network. We also halve runtime memory



Figure 4: Qualitative results on hands+body. **1st row**: Results on VR sessions on Elixir/Hand Physics Lab. **Row 2**: Results on SIGNUM [D] dataset **Row 3**: Results from the Leeds Sport Dataset []. **Row 4**: Pose estimation results in YouTube clips.

requirements and training time since we only need to perform it for left hands.

Training of the ensemble: We use the Tensorflow 2 [13] deep-learning framework, the RMSProp optimizer with batch sizes of 128 samples, learning rate of 0.00013, $e = 10^{-6}$ (see [13]) and train each encoder for 45 epochs (see [13]). The loss function is mean squared error (MSE) between ground truth and each 3D joint rotation prediction. Our training order follows the hand joint hierarchy order. We start from the wrist and progressively go through each finger from pointer to pinky with the thumb trained last. Each encoder for each d.o.f. of each joint is trained separately but initialized using its parent's network weights in an attempt to transfer knowledge previously acquired on our training session and to avoid trying to win the randomized initialization "lottery ticket" [52] for each and every encoder. We terminate training if loss improvement is less than 0.001 in 5 or more consecutive training selecting the weights that achieved minimum loss in the training session regardless of chronological order.

Weakly-Supervised training dataset generation: In contrast to the baseline [\Box_2 , \Box_3] that employed BVH MOCAP [\Box_3] from the Carnegie Mellon dataset [\Box_3] we could not identify a similar BVH source for hands. In addition, research shift towards weakly/semi supervised methods [\Box_1 , \Box_3 , Ξ_3], the need for dataset filtering [\Box_3] and works that use MOCAP data just to extract rotation limits [\Box_1] prompted a weakly supervised approach. Studying the literature we find anatomically correct hand [\Box_3] and wrist [Ξ_3] kinematic models which we enforce in our random pose generator. Our baseline [\Box_3] relies on 2.2*M* samples for each ensemble. This amounts to 8.8*M* samples covering all orientations. After several tests we end-up with the following randomization scheme. All added hand poses feature randomized 3D positions and rotations. We add 2*M* fully random poses, 200*K* with all finger d.o.f. set to zero and 600*K* with a naturally open hand and no finger movement. Finally each finger gets 400*K* uniform random poses with all other fingers open and 400*K* where the rest of the hand is closed. This randomization scheme emphasizes learning of 3D orientation, ensures all fingers receive equal training samples and yields a total of $\approx 7.2M$ samples, a number close both to the baseline sample number and our technical CPU/GPU memory limitations.

Pose fine-tuning: We adopt the Hierarchical Coordinate Descent (HCD) [\square] as the final refining step of our pipeline. In comparison to other solvers like CERES [\square] or FABRIK [\square], HCD is purposefully built to deal with the sporadic conditionally independent errors encountered in the output of our NN ensemble. We define 6 kinematic chains *C* with *C*₂ to *C*₆



Figure 5: Left: Hand pose estimation accuracy on STB [22] and RHD [1116] for increments of synthetic Gaussian noise. We observe that the impact of 2D noise is less pronounced compared to the impact of non-temporally cohesive poses. **Right:** Proposed body pose method accuracy on H36M [12] for increments of synthetic Gaussian noise vs baseline [17]. We observe consistent accuracy improvements despite the lack of an orientation classifier.

corresponding to each finger. In contrast to the baseline body [$[\Box]$] estimation that always executes on the same chains, we implement a logic switch that alters C_1 from standalone to holistic operation depending on occlusions. In case of an upper body present in our observations, C_1 begins at the shoulder joint including elbow, wrist and finger metacarpal bones as end points. If 2D observations carry no body information C_1 begins at the wrist extracting an absolute 3D orientation of the hand. Thus, C_1 acts as a mediator chain, propagating knowledge of the upper body to hands and vice versa. Regardless of C_1 , HCD concurrency is not affected since e.g. hand optimization has no bearing in lower body results. Dependencies only rise on upper body+hand execution but we are still able to execute all HCD sessions in parallel syncing solution updates on each iteration for 35 HCD iterations of 30 epochs each.

We introduce two novel sets of optimization limits to enhance HCD [2]. The first set enforces mechanical hand limits as hard constraints across all updates. This guarantees output in the range of valid rotations. The second set increases HCD efficiency. We often observe exploding gradients that needlessly occupy CPU time since despite error diverging HCD execution continues. To combat this, we note loss/MAE achieved for each NN hand encoder which informs us about the average expected HCD correction. Using training MAE as a delta limit for each HCD epoch, gradient explosions are suppressed improving efficiency.

4 **Experiments**

For the quantitative assessment of our method we relied on well-established standalone hand and body pose datasets. This is possible due to our formulation that can addresses them in isolation. For qualitative evaluation we use a number of diverse use cases. We record VR sessions in retail applications to ensure realistic scenarios, we showcase our method on the SIGNUM dataset [12] that has a variety of distinct sign language gestures. We attempt body+hand pose estimation at the Leeds Sports Dataset [12] since hand pose has not been considered at it. Finally, we capture challenging performances "in-the-wild" from YouTube. **Quantitative hand pose estimation experiments:** Our method's compositionality allows hand tracking execution isolated from the rest of the pipeline. This allows experiments that can compare our method with standalone 3D hand pose estimators. The STB [12] dataset features a variety of hand poses recorded using a depth camera with annotated ground truth. Position of the root joint and global scale is aligned to ground truth in the literature [110], [116] a practice we also follow for comparable results. A dataset issue is that it features a palm joint instead of a wrist. Most hand models, including our own, feature wrist joints since armatures mimic the natural bone structure of hands. Following the literature, we handle this

QAMMAZ, ARGYROS: TOWARDS HOLISTIC REAL-TIME HUMAN 3D POSE ESTIMATION 9

Input	Dir	Dis	Eat	Gre	Pho	Pos	Pur	Sit	Smo	Pho	Wai	Wal	Dog	WaT	Sit.	Avg
Our Method	71	78	107	86	98	79	113	121	99	98	84	100	122	104	138	99
MNET20 [🗳]	76	93	107	88	102	86	143	147	104	111	97	96	143	88	162	108
MNET19 [135	140	145	143	153	137	174	215	156	150	151	156	166	134	246	160

Table 1: Comparison of our method with the baseline on body pose estimation. Methods are trained on CMU and tested using H36M Blind Protocol 1. All numbers are in millimeters.

							[Our						21		
160	119	118	116	113	108	108	107	101	99	93	88	88	88	82	80	72	40

Table 2: Comparison of 3D body estimators on H36M Protocol 1 (Method / MPJPE in mm).

palm/wrist dislocation via linear approximation between the wrist and metacarpal [2], [1]] and do not consider the wrist joint in error calculations. We achieve a mean average error (MAE) in the sub 10mm zone. STB dataset poses are consecutive and use only the left hand. We test on the RHD dataset [1116] that has no temporal continuity and features both hands to gain more insight on our method accuracy. Our method manages to retain a MAE of 25.37mm indicative of the expected accuracy and rate of recovery in cases of abrupt multi-frame hand occlusions. The Percentage of Correct Keypoints (PCK) curves are reported in Figure 5 and comparison with other methods is summarized in Table 3. Despite an average 3.22 mm higher STB error than SOTA we manage an excellent accuracy / performance ratio since the only other real-time method of Table 3 is [116], it uses GPGPUs, and we achieve similar frame-rates to it on CPU-only execution while also computing the body pose.

Quantitative body pose estimation experiments: We "back propagate" our methodological improvements for hand pose estimation to the body estimation task. Evaluation is performed on the H36M dataset [13] through mean per joint estimation error (MPJPE) after Procrustes alignment [13] compared to ground truth. We use the Blind P1 [12], [13] protocol to have results that are directly comparable with [12], [13]. Table 1 offers an accuracy breakdown for H36M exhibiting improvements across most tasks. Results reveal an error reduction of 9% relative to the baseline [13] despite the 51% frame-rate increase from 70Hz [13] to 106Hz. Table 2 shows our accuracy compared to competing body pose estimation methods.

Ablation study: We perform experiments to examine the 3D hand pose estimation accuracy effects of HCD, training sample size, eNSRM vs NSRM [23] descriptors, SELU [53] vs SWISH [24] activation functions and multiple d.o.f. regression via the same encoder. Due to space limitations, the detailed results have been published along with the project source code [3]. We observe (a) the importance of HCD, particularly in temporarily cohesive input like STB [24], (b) that larger randomized sample sizes improve resulting trained encoders especially using the more complex eNSRM descriptor, (c) that SWISH and SELU perform similarly and (d) that while regressing multiple d.o.f. from each encoder is feasible, it quickly degrades output accuracy as more d.o.f. share the capacity of a single encoder.

Qualitative results in VR scenarios: We employ two retail Oculus Quest VR applications, Elixir and Hand Physics Lab (HPL), for qualitative assessment of our method. They are

	[Our	[12]				[[22]	[[]]
RHD	30.42	25.37	20.89	19.95	19.73	17.11	15.77	15.61	13.88
STB	8.68	9.93	7.95	8.66	8.56	7.27		6.93	6.71

Table 3: Comparison of 3D hand estimators tested on RHD/STB (Method / MPJPE in mm).

	1 Output	2 Outputs	3 Outputs	5 Outputs
STB	18.28	19.80	25.58	27.03

ł

Table 4: Multiple output regression experiments via a single encoder on STB [22] (Method / MPJPE in mm). We do not perform the HCD step to measure the unskewed NN result. We observe deteriorating accuracy as more outputs are regressed using the same encoder.

both state of the art in terms of the provided onboard VR hand tracking experience and feature a variety of interesting to track in-game tasks and interactions. Elixir takes place in a room-sized space requiring lateral movement while HPL doesn't since all in-game objects are within arms reach. Our camera faces the user in the initial orientation of the application when it first loads. Testing both scenarios allows us to study our method while controlling for body pose variations. We observe high fidelity 3D BVH skeletons throughout the activities.

Qualitative results in sports: Leeds Sport Dataset [12] consists of standalone frames depicting various sport activities with no temporal, subject or activity cohesion. It bears similarities to RHD [1007] albeit targeting bodies with non synthetic images. Our method accommodates this challenging task extracting BVH skeletons including hands for the first time.

Qualitative results in sign language: The SIGNUM dataset [**D**] is a multi person monocular RGB dataset designed for sign language system training. We use it to qualitatively test our hand+body pose estimation ensemble with its fast, complex and challenging sign language gestures. Our method manages to generalize across multiple users of different genders and body types and we observe high fidelity 3D pose capture. The lower-body is constantly cropped out of the input feed, testing the ensemble occlusion tolerance properties [**D**].

Qualitative results in in-the-wild Youtube videos: We browse YouTube and collect videos that contain dancing and instrument performances that show interesting body and hand motions. We proceed to successfully track them despite the videos exhibiting unknown camera setups, rapid camera changes, rapid motions, self-occlusions and motion blur.

Sample qualitative results are illustrated in Figure 4. More results appear in the supplementary video [2] accompanying this paper.

5 Discussion

We successfully adapted a body pose estimation method to tackle the hand pose estimation problem. The proposed methodological novelties not only allowed us to achieve this goal, but yielded improved results on the original body pose estimation problem. The combined human+body pose estimation ensemble we propose has many interesting properties. Hierarchies that are occluded or very far can be dropped from execution without impacting visible parts, output is directly derived in the popular BVH file format making the method accessible to a wide audience and its resource efficiency allows it to be deployed on generic PCs matching most RGB webcam frame-rates. We performed quantitative evaluation of the proposed method in standard hands-only and bodies-only datasets. We also tested its performance in a number of use cases including VR, sports tracking, sign language and other challenging scenarios. Future research could extend the ensemble with a facial encoder. Facial controls are already present in our BVH skeleton, albeit currently not populated. This could extend the method to a real-time body+hands+face 3D estimation stack from monocular RGB.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Quadro P6000 GPU used for the execution of this research. The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I) under the "1st Call for H.F.R.I Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment" Project I.C.Humans, Number: 91

References

- [1] A. Qammaz. Mocapnet 3 ablation study results, 2021. URL https: //github.com/FORTH-ModelBasedTracker/MocapNET/tree/ master/doc/ablation_bmvc21. [Online; accessed 21-Oct-2021].
- [2] A. Qammaz. Mocapnet 3 supplementary video, 2021. URL https://youtu.be/ aaLOSY_p6Zc. [Online; accessed 21-Oct-2021].
- [3] Timo et.al. Aho. Multi-target regression with rule ensembles. *Journal of Machine Learning Research*, 13(8), 2012.
- [4] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of IEEE CVPR*, pages 1446–1455, 2015.
- [5] Andreas Aristidou and Joan Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011.
- [6] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning, pages 37–49. JMLR Workshop and Conference Proceedings, 2012.
- [7] Pierre Baldi and Peter J Sadowski. Understanding dropout. In Advances in neural information processing systems, pages 2814–2822, 2013.
- [8] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.
- [9] Blender Community. *Blender a 3D modeling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2019. URL blender.org/.
- [10] Liefeng Bo and Cristian Sminchisescu. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28, 2010.
- [11] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.
- [12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.
- [13] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017.

- [14] François Chollet et al. Keras. https://keras.rstudio.com/reference/ fit.html, 2015.
- [15] François Chollet et al. Keras model checkpoints, 2015. URL https://keras. io/callbacks/#modelcheckpoint. [Online; accessed 8-April-2019].
- [16] François Chollet et al. Keras. https://keras.io/api/optimizers/ rmsprop/, 2015.
- [17] Salvador Cobos, Manuel Ferre, M.A. Sanchez Uran, Javier Ortego, and Cesar Pena. Efficient human hand kinematics for manipulation tasks. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2246–2251, 2008. doi: 10.1109/IROS.2008.4651053.
- [18] Yu et.al. Du. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *ECCV*, pages 20–36. Springer, 2016.
- [19] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.
- [20] Dushyant Mehta et.al. Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics, 36(4), 2017. doi: 10.1145/3072959. 3073596. URL http://gvv.mpi-inf.mpg.de/projects/VNect/.
- [21] Georgios Pavlakos et.al. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.
- [22] Gül Varol et.al. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018.
- [23] Hanbyul Joo et.al. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.
- [24] Jiawei Zhang et.al. 3d hand pose tracking and estimation using stereo matching. *arXiv* preprint arXiv:1610.07214, 2016.
- [25] Marta Sanzari et.al. Bayesian image based 3d pose estimation. In *European conference on computer vision*, pages 566–582. Springer, 2016.
- [26] Sameer Agarwal et.al. Ceres solver. http://ceres-solver.org.
- [27] Simone Bianco et.al. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018.
- [28] Yucheng Chen et.al. Monocular human pose estimation: A survey of deep learningbased methods. *Computer Vision and Image Understanding*, page 102897, 2020.
- [29] Zhe Cao et.al. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In arXiv preprint arXiv:1812.08008, 2018.
- [30] Shachar Fleishman, Mark Kliger, Alon Lerner, and Gershom Kutliroff. Icpik: Inverse kinematics based articulated-icp. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 2015.

- [31] Roger Fletcher. Practical methods of optimization. John Wiley & Sons, 2013.
- [32] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.
- [33] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.
- [34] Jiajun Gu, Zhiyong Wang, Wanli Ouyang, Jiafeng Li, Li Zhuo, et al. 3d hand pose estimation with disentangled cross-modal latent space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 391–400, 2020.
- [35] Yuliang Guo, Zhong Li, Zekun Li, Xiangyu Du, Shuxue Quan, and Yi Xu. Pop-net: Pose over parts network for multi-person 3d pose estimation from a depth image. arXiv preprint arXiv:2012.06734, 2020.
- [36] B. Hahne. The daz-friendly bvh release of cmu motion capture database, 2010. URL https://sites.google.com/a/cgspeed.com/cgspeed/ motion-capture/daz-friendly-release. Accessed: 2018-10-05.
- [37] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. ACM Transactions on Graphics (TOG), 39(4):87–1, 2020.
- [38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [39] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.
- [40] HTC. Htc vive tracker, 2020. URL https://www.vive.com/us/ accessory/vive-tracker/. [Online; accessed 16-Feburary-2021].
- [41] HTC. Vrchat full-body tracking guide, 2020. URL https://docs.vrchat. com/docs/full-body-tracking. [Online; accessed 16-Feburary-2021].
- [42] Gao et.al. Huang. Densely connected convolutional networks. In *Proceedings of the IEEE CVPR*, pages 4700–4708, 2017.
- [43] Dong-Hyun et.al Hwang. Lightweight 3d human pose estimation network training using teacher-student learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 479–488, 2020.
- [44] Autodesk inc. Motionbuilder 3d character animation, 2019. URL https://www.autodesk.com/products/motionbuilder/. Accessed: 2019-10-05.
- [45] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36 (7):1325–1339, 2014.
- [46] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

- [47] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Citeseer, 2010.
- [48] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference* on computer vision and pattern recognition, pages 8320–8329, 2018.
- [49] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-toend recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.
- [50] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [51] Günter et.al. Klambauer. Self-normalizing neural networks. In Advances in neural information processing systems, pages 971–980, 2017.
- [52] Ilya Kostrikov and Juergen Gall. Depth sweep regression forests for estimating 3d human pose from images. In *BMVC*, volume 1, page 5, 2014.
- [53] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [55] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [56] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.
- [57] Rui Li, Zhenyu Liu, and Jianrong Tan. A survey on 3d hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition*, 93:251–272, 2019.
- [58] Hui Liang, Junsong Yuan, Daniel Thalmann, and Zhengyou Zhang. Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization. *The Visual Computer*, 29(6):837–848, 2013.
- [59] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG), 34(6):248, 2015.
- [60] MakeHuman Community. Makehuman, 2019. URL http://www.makehumancommunity.org/. [Online; accessed 8-April-2019].
- [61] Alexandros Makris, Nikolaos Kyriazis, and Antonis A Argyros. Hierarchical particle filtering for 3d hand tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 8–17, 2015.
- [62] Dushyant et. al. Mehta. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

- [63] Dushyant et. al. Mehta. Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1907.00837*, 2019.
- [64] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 184–184, 2013.
- [65] Maddock Meredith, Steve Maddock, et al. Motion capture file formats explained. *Department of Computer Science, University of Sheffield*, 211:241–244, 2001.
- [66] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 4, 2019.
- [67] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2498–2507. JMLR. org, 2017.
- [68] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. *arXiv preprint arXiv:2008.09309*, 2020.
- [69] Natalia Neverova, Christian Wolf, Florian Nebout, and Graham W Taylor. Hand pose estimation through semi-supervised and weakly-supervised learning. *Computer Vision and Image Understanding*, 164:56–67, 2017.
- [70] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011.
- [71] Xue Bin et.al. Peng. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM TOG, 37(4):143, 2018.
- [72] Ammar Qammaz and Antonis A Argyros. Mocapnet: Ensemble of snn encoders for 3d human pose estimation in rgb images. In *BMVC*, page 46, 2019.
- [73] Ammar Qammaz and Antonis A. Argyros. Occlusion-tolerant and personalized 3d human pose estimation in rgb images. In *IEEE International Conference on Pattern Recognition (ICPR 2020)*, January 2021.
- [74] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.
- [75] James M Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European conference on computer vision*, pages 35–46. Springer, 1994.
- [76] Konstantinos Roditakis and Antonis A Argyros. Quantifying the effect of a colored glove in the 3d tracking of a human hand. In *ICVS*, pages 404–414, Copenhagen, Denmark, July 2015. Springer. doi: 10.1007/978-3-319-20904-3_36.
- [77] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localizationclassification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017.

- [78] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multiperson 2D and 3D Pose Detection in Natural Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [79] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics (ToG), 36 (6):1–17, 2017.
- [80] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.
- [81] Matthias Schröder, Jonathan Maycock, Helge Ritter, and Mario Botsch. Real-time hand tracking using synergistic inverse kinematics. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 5447–5454. IEEE, 2014.
- [82] Mingyi et.al. Shi. Motionet: 3d human motion reconstruction from monocular video with skeleton consistency. ACM Transactions on Graphics (TOG), 40(1):1–15, 2020.
- [83] M Soubeyrand, B Assabah, M Bégin, E Laemmel, A Dos Santos, and M Crézé. Pronation and supination of the hand: anatomy and biomechanics. *Hand Surgery and Rehabilitation*, 36(1):2–11, 2017.
- [84] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.
- [85] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. *arXiv preprint arXiv:2003.09282*, 8, 2020.
- [86] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.
- [87] Andreas et. al. Stergioulas. 3d hand pose estimation via aligned latent space injection and kinematic losses. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 1730–1739, 2021.
- [88] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 824–832, 2015.
- [89] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision* (ECCV), pages 529–545, 2018.
- [90] Poonam Suryanarayan, Anbumani Subramanian, and Dinesh Mandalapu. Dynamic hand pose recognition using depth data. In 2010 20th International Conference on Pattern Recognition, pages 3105–3108. IEEE, 2010.
- [91] Vince Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. 2018.

- [92] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Fusing 2d uncertainty and 3d cues for monocular body pose estimation. *arXiv preprint arXiv:1611.05708*, 2(3), 2016.
- [93] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.
- [94] Thomas Theodoridis, Theocharis Chatzis, Vassilios Solachidis, Kosmas Dimitropoulos, and Petros Daras. Cross-modal variational alignment of latent spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 960–961, 2020.
- [95] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference* on Computer Vision and Pattern Recognition, pages 2500–2509, 2017.
- [96] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.
- [97] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.
- [98] Carnegie Mellon University. Cmu graphics lab motion capture database. http: //mocap.cs.cmu.edu/, 2003. Accessed: 2017-06-01.
- [99] Ulrich Von Agris and Karl-Friedrich Kraiss. Towards a video corpus for signerindependent continuous sign language recognition. *Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal, May*, 11, 2007.
- [100] Wikipedia contributors. Euclidean distance matrix wikipedia, 2018. URL https://en.wikipedia.org/w/index.php?title=Euclidean_ distance_matrix&oldid=862708912. [Online; accessed 8-April-2019].
- [101] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10974, 2019.
- [102] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision* and Pattern Recognition, pages 9877–9886, 2019.
- [103] H. et. al. Yasin. A dual-source approach for 3d pose estimation from a single image. In *IEEE CVPR*, 2016.
- [104] Xiaowei et al. Zhou. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on CVPR*, pages 4966– 4975, 2016.
- [105] Xingyi et. al. Zhou. Deep kinematic pose regression. In *ECCV*, pages 186–201. Springer, 2016.
- [106] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE ICCV*, pages 4903–4911, 2017.