TravelNet: Self-supervised Physically Plausible Hand Motion Learning from Monocular Color Images

Zimeng Zhao Xi Zhao Yangang Wang*

Southeast University, China

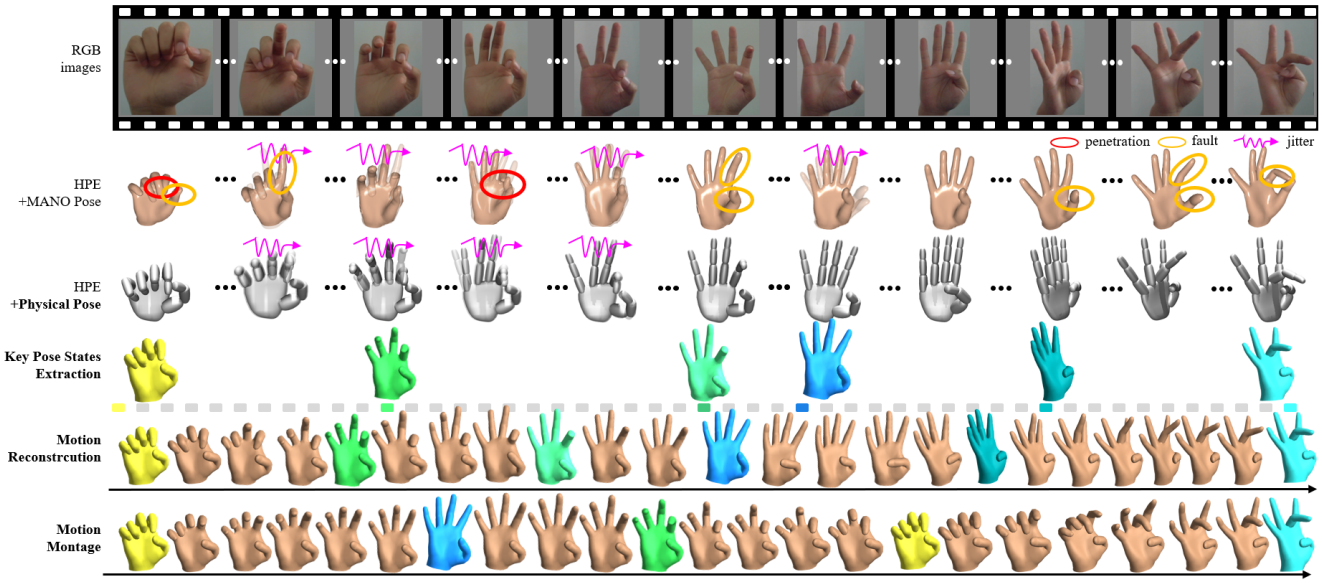


Figure 1. **Physically plausible hand motion learning.** From a series of monocular color images (row 1), adapting our physical pose representation to the hand pose estimator can get fewer penetrations and higher accuracy (row 2 and 3). **TravelNet** learns to travel in this physical pose manifold from given key pose states (row 4). It can not only reconstruct the original motion (row 5) but also generate various montages based on it (row 6). Due to the lack of RGB sequence data and key pose state annotations, a *self-supervised* learning paradigm is adopted, and its training motion data is generated by a physics engine with our pose state archive.

Abstract

*This paper aims to reconstruct physically plausible hand motion from monocular color images. Existing frame-by-frame estimating approaches can not guarantee the physical plausibility (*e.g. *penetration, jittering) directly. In this paper, we embed physical constraints on the per-frame estimated motions in both spatial and temporal space. Our key idea is to adopt a self-supervised learning strategy to train a novel encoder-decoder, named TravelNet, whose training motion data is prepared by the physics engine using discrete pose states. TravelNet captures key pose states from hand motion sequences as compact motion descriptors, inspired by the concept of keyframes in animation. Finally, it manages to extract those key states out of perturbations without manual annotations, and reconstruct the motions preserving details and physical plausibility. In the experiments, we show that the outputs of the TravelNet contain both finger synergism and time consistency. Through the proposed framework, hand motions can be accurately reconstructed and flexibly re-edited, which is superior to the state-of-the-art methods.*

1. Introduction

Plausible human hand motions are of paramount importance in many applications. In VR/AR, reconstructing plausible hand motion facilitates closer interaction. In manipulation planning, designing a plausible hand motion makes a bionic hand more intelligent to assist the disabled. Con-

*Corresponding author. E-mail: yangangwang@seu.edu.cn. All the authors from Southeast University are affiliated with the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China.

ventionally, high fidelity hand motions are collected by data gloves [1, 2] or specialized hardware devices [4], though they are expensive and cumbersome.

In recent years, with extensive hand pose datasets, deep learning has witnessed the rapid progress of hand pose estimation from depth images [35, 30] and monocular color images [60, 20, 58]. Theoretically, most of them can be extended to estimate motions with appropriate temporal or recurrent modules [18, 5] and plenty of motion sequences. However, due to the diversity and composability of hand motions, it is challenging to prepare and label those sequences offline.

To solve the difficulties of hand motion collection, a few methods [28, 34] attempted to refine the results of per-frame estimation by avoiding high-frequency jittering. Recently, Yang *et al*. [51] proposed a synthesis method that generated a motion sequence by performing linear interpolation among sample states in the dataset and finds the nearest pose instances in the dataset. However, this product might not be physically plausible (shown as Fig. 7) because it can not guarantee the existence of a dense and continuous path composed of linear transients between any two pose states.

There are two main challenges to learn physically plausible hand motions. The first one is that it is laborious to prepare sufficiently diverse as well as plausible motion data. The second one is that it is difficult to distinguish the informative pose states from the jitter ones without any annotations. In this paper, we focus on the problem of learning physically plausible hand motions from a set of discrete hand pose states, which are estimated from monocular color images. Our key idea is to train a novel encoder-decoder network, named TravelNet, whose training motion data is prepared with the help of a physics engine. TravelNet manages to find key pose states out of the perturbation and reconstruct the motion that preserves details and physical plausibility.

Nevertheless, it is hard to define and annotate key pose states in a motion sequence. To solve the obstacles, we propose a novel self-supervised paradigm to perform training. Specifically, we ensure that the embedded space outputted by the encoder of TravelNet remains in the same pose manifold as the input space (as shown in Fig. 3). The decoder is first trained to output the motion with discrete pose states, where hand motions are guided by an inverse dynamic solver from the physical engine. This well-trained decoder assists the training of the decoder for the next step. Finally, the encoder and decoder are combined with fine-tuning on a limited number of real hand motion sequences as a domain adaption strategy.

To ensure the physical plausibility of hand motion training data, we build a hand model incorporating physical constraints in the physical engine [3] to detect collisions and calculate inverse dynamics effortlessly. A pose state that passes the penetration validation is called a physical pose state because it is physically plausible. We then map extensive hand poses from multi-modal hand datasets [53, 54, 61, 13, 57, 40, 46, 25] to the above physical pose states, which provide abundant primitives and prior knowledge for generating plausible motion sequences.

The main contributions of this work are summarized as follows.

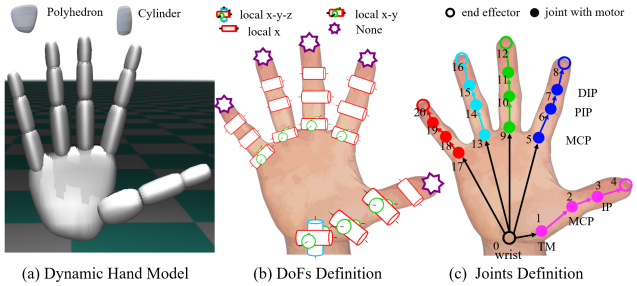
- A novel learning paradigm that can extract key pose states robustly and reconstruct the hand motion in a self-supervised manner;
- A physical pose bound to a dynamic hand model is adopted as the compact descriptor of hand motion;
- An archive containing 2.5M physical hand poses are created for plausible motion generation by an inverse dynamic solver.

The dataset and codes will be publicly available at https://www.yangangwang.com.

2. Related Work

Monocular RGB Hand Pose Estimation. It is a hot research topic to learn 3D hand pose from a single RGB image. Some pioneers [60, 20, 42, 31] directly predicted the joint 3D coordinates. Later work [55, 7, 12, 58], however, tends to rely more on a popular rigged model, MANO [39] to estimate its pose parameters (axis angle of each joint) from the image. One of the painful problems in MANO is that the invalid pose parameters may cause the penetration of the deformed mesh surface. Because the computation of penetration is time-consuming, it is more used in the offline optimization [41] or training stage [17, 29] than in the feedforward of the network. Furthermore, although the frame-by-frame estimation in the offline dataset has been improved, severe jitter is included in the results when applying it to a whole motion sequence.

Motion Synthesis. Motion synthesis has the potential to provide realistic data for detection or segmentation tasks in the CV field. Existing methods use game engines [60] and deep generative models [31] to acquire synthetic datasets. They focus more on the augment at pixel level instead of the diversity of pose states. There exist continuous sequences in [53] recorded as depth maps, and RGB-based tasks cannot directly use them as training data. Using [53] as a retrieval database, Yang *et al*. [51] proposed a hand motion synthetic scheme that uses the method of linear interpolation and the nearest approximate neighbor samples. Yamamoto *et al*. [50] utilized the jumps in the score to find the keyframes in the hand motion when playing the piano, whose motion acquisition relies on the modelling of hands and piano keyboard and cannot be generalized to a wider range of applications. Besides, [16] utilized GAN-based [14] network to synthesize the motion recurrently, and [36] trained a control



(a) Dynamic Hand Model (b) DoFs Definition (c) Joints Definition

Figure 2. **Physical hand model.** Our hand model consists of rigid body segments connected by joints. Penetrations can be tackled by a collision proxy with this model. Elementary motions can be generated by an inverse dynamic solver when the start and end states are determined.

strategy for the human body motion in the manner of reinforcement learning. A fixed-length motion sequence might contain an *elementary* (*e.g.* count slowly from 1 to 2 using fingers) or a *composite* motion (*e.g.* count quickly from 1 to 9 using fingers). Previous methods [19, 22] performed motion modelling without distinction between the two cases. However, we argue that only the elementary motion can be determined by a pair of endpoints. A composite motion can be reconstructed as the original one only if those key states that characterize the origin are not destroyed by noise.

Motion Keyframe Extraction. Our key pose states describe a hand motion similar to the keyframes used to summarize video content. Given an image sequence, generalized keyframe extraction [48, 26] uses changes in optical flow and SIFT features as the criteria. For motion sequences, some literature [15, 24] adopted the 2D pose (key-points) to describe each frame, others[59, 38, 44] used unsupervised clustering methods to find keyframes. As a learning-based clustering method, adaptive mean-shift with learnable bandwidth was applied to mesh rigging joints proposal by Xu *et al*. [49]. We follow a similar strategy to extract key pose states in a motion sequence.

3. Hand Model and Representation

Physical Hand Model. Our articulated hand is a rigid body adapted in the physics engine [3] as shown in Fig. 2 and Fig. 3 (A). It is created by approximating MANO [39] mesh as 16 polyhedral segments and assigning 21 degrees of freedom (DoFs), which is defined as [10]. Corresponding physical properties (mass, friction, *etc*.) are estimated according to the volume and boundary of each polyhedron. With this model, both collision detection and inverse dynamic can be solved by corresponding proxies or solvers in the physics engine [3].

Pose and Motion Formulation. The hand pose state representation $\boldsymbol{\theta} \in \mathbb{R}^{21}$ used in TravelNet is bounded to each DoF of our articulated hand model. Global transformation $\mathbf{R}, \boldsymbol{t}$ is not considered in TravelNet, *i.e*., all the pose states are aligned in a canonical space. Each dimension of $\boldsymbol{\theta}$ is constrained to $[-0.5\pi, 0.5\pi]$. We regard a hand motion as a set of pose states, which is denoted as Θ_N. The subscript N is the number of pose states. It is noted that Θ may be continuous or discontinuous in time. For example, the corresponding key pose states set Θ_K is a subset of a continuous Θ_N, which is discontinuous and contains K states selected from Θ_N. And other $(N-K)$ trivial states in Θ_N is widely named as in-between pose states.

4. Pose State Archive Preparation

We construct a pose state archive from existing datasets. During the later TravelNet training phase, the inverse dynamic solver would randomly select a state from the archive in order to generate the associated motion data.

Physical Pose Estimator. To consolidate the pose state knowledge as much as possible, we first establish a physical pose estimator that translates the discrete pose states in the existing RGB image datasets under our representation. It also decouples the global and local transformation $\boldsymbol{\theta}$ from the 3D joints location in the estimation process, leading all poses in a canonical space. More details of this part are provided in the **Sup. Mat** .

Pose Archive. By using all or part of the pipeline of the pose estimator described above, the pose state priors in both RGB and depth images can be unified to be our physical pose states. As a result, an off-line pose state archive containing 2.5 M instances shown in Tab. 1 is constructed. Each discrete pose state is stored as a data mapping between the 3D joints locations and corresponding $\boldsymbol{\theta}$.

Name	No. Frames	No. Subjects	No. Sequences	Modality
STB [54]	18,000	1	12	RGB MoCap
MHP [13]	76,375	9	21	RGB MoCap
Frei [61]	130,240	24	-	RGB MoCap
Hand 3D Studio [57]	42,960	-	-	RGB MoCap
CMU-MPII [40]	1,445	-	-	RGB in the wild
OneHand10K [46]	11,703	-	-	RGB in the wild
Halpe [25]	47,776	-	-	RGB in the wild
BigHand2.2M [53]	2.2 M	10	99	Depth Automatic
Pose Archive	2.5 M	Unified	132	-

Table 1. **Pose Archive Components.** Multimodal data is introduced in our archive. Only sequences in [13, 53] are used for TravelNet fine-tuning. Other data is regarded as a state set from which inverse dynamic solver can randomly select a subset to generate various motions during TravelNet training.

5. TravelNet

TravelNet learns the motions from the perspective of the key pose states. The whole pipeline of the proposed TravelNet is shown in Fig. 3. Although TravelNet is designed as a deep encoder-decoder architecture, we can not train it as a traditional auto-encoder due to the lack of RGB sequence data and key pose state annotations. Alternatively, we propose a novel three-step self-supervised paradigm. The details are described in the following.

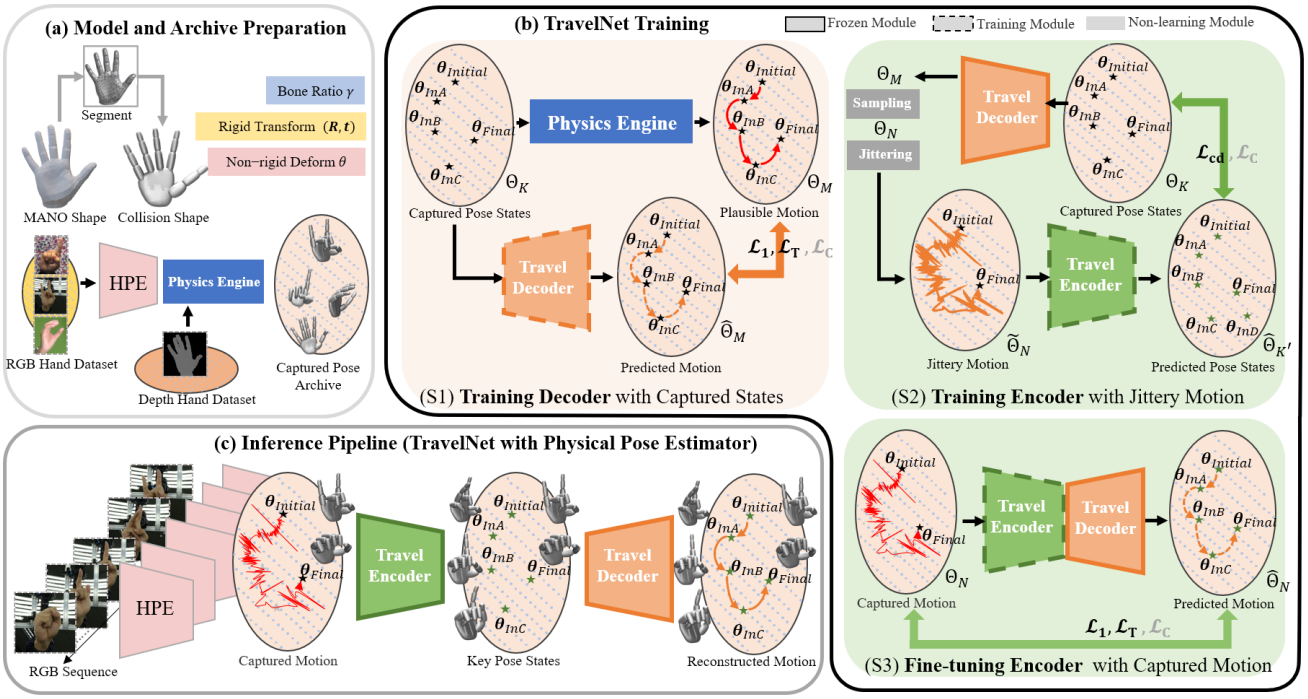


Figure 3. **Overview of TravelNet**. (a) Pose representation. The details are discussed in Sec. 3, and training data generation in Sec. 4. (b) Training phase. It is designed as three-step self-supervised paradigm described in Sec. 5.1 to 5.3. (c) Inference phase. It finally can extract the key pose states and reconstruct the hand motion without the dependence of physics engine.

5.1. Decoder Trained with Captured States

An inverse dynamic solver in the physical engine can optimize a plausible motion between arbitrary two end-points with time, energy and collision constrains. When considering multiple captured pose states Θ_K in the archive, this process can be performed recurrently to each adjacent data pair $\{\boldsymbol{\theta}_{k_i}, \boldsymbol{\theta}_{k_{i+1}}\}, k_i \in K$ to get elementary segments and finally concatenate to be a composite one Θ_M. Our decoder $\mathcal{D}_{trv}$ imitates this process by:

$$\hat{\Theta}_M = \left[\mathcal{D}_{trv}(\boldsymbol{\theta}_{k_1}, \boldsymbol{\theta}_{k_2}) \,\vdots\, \cdots \,\vdots\, \mathcal{D}_{trv}(\boldsymbol{\theta}_{k_{K-1}}, \boldsymbol{\theta}_{k_K}) \right] \tag{1}$$

To guarantee a fixed length of Θ_M, each elementary segment in Θ_M is farthest-point sampled [37] or replication-padded to $D = 1024$ pose states, leading to $M = (K-1)D$. The overall loss of the decoder consists of three parts:

$$\mathcal{L}_{\text{S1}} = \mathcal{L}_1(\hat{\Theta}_M, \Theta_M) + w_T \mathcal{L}_{\text{T}}(\hat{\Theta}_M) + w_C \mathcal{L}_{\text{C}}(\hat{\Theta}_M) \tag{2}$$

where $w_T = 0.6, w_C = 0.05$ in all of our experiments.

The first term $\mathcal{L}_1$ performs the supervision between $\hat{\Theta}_M$ predicted by the decoder and Θ_M obtained by inverse dynamics:

$$\mathcal{L}_1(\hat{\Theta}_M, \Theta_M) = \|\hat{\Theta}_M - \Theta_M\|_1. \tag{3}$$

The second term $\mathcal{L}_{\text{T}}$ balances the consistency between piece-wise smoothness in each segment and global smoothness:

$$\mathcal{L}_{\text{T}}(\Theta_M) = \sum_{i=1}^{M-1} \lambda_i \mathcal{L}_1(\boldsymbol{\theta}_{i+1}, \boldsymbol{\theta}_i), \tag{4}$$

$\lambda_i = 1.0$ when $\boldsymbol{\theta}_i \in \Theta_K$, and $\lambda_i = 0.75$ for other in-between states.

We also introduce a GMM-based collision penalty [32] $\mathcal{L}_{\text{C}}$ though collision has been implicitly included by the data itself:

$$\mathcal{L}_{\text{C}}(\boldsymbol{\theta}) = \sum_{p=1}^{N_C} \sum_{q=p+1}^{N_C} \int_{\mathbb{R}^3} G_p(\boldsymbol{x}; \boldsymbol{\theta}) \cdot G_q(\boldsymbol{x}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{x}, \tag{5}$$

where G_p, G_q denote the Gaussian collision proxies depend on state $\boldsymbol{\theta}$. N_C denotes the proxies number. In computation, only the MANO vertex positions are taken into account in $\mathrm{d}\boldsymbol{x}$. For a motion sequence, $\mathcal{L}_{\text{C}}(\Theta_M) = \sum_{\boldsymbol{\theta} \in \Theta_M} \mathcal{L}_{\text{C}}(\boldsymbol{\theta})$.

5.2. Encoder Trained with Jittery Motions

Due to the lack of the key pose states annotations in real captured motions, the well-trained decoder in Sec. 5.1 is used to assist the encoder's learning. Different from a composite motion Θ_M (treated as ground truth in this step) generated by $\mathcal{D}_{trv}$, a captured motion often has three cases including (I) a variable length, (II) an unfixed number of in-between states in each segment, and (III) high-frequency jitter. So we further augment Θ_M by random sampling to be Θ_N and then random jittering to be $\tilde{\Theta}_N$. It is noted neither

operation changes the identity of the key pose states, which means $\Theta_M, \Theta_N, \tilde{\Theta}_N$ share the same Θ_K.

Sampling. To disturb each segment in Θ_M with unfixed in-between poses, $\{d_i\}_{i=1}^{K-1}$, i.i.d $d_i \sim \mathcal{U}(0, D/2)$ are generated to determine in-between poses number that should be deleted in each elementary segments. This makes Θ_M to be Θ_N, where $N = M - \sum_{i=1}^{K-1} d_i$. Since the full convolution and clustering block will be adopted later, neither the length of Θ_K to generate them nor Θ_N is necessary to be fixed.

Jittering. To add jitter to the motion, a series of jitter masks $\boldsymbol{M_i}(\lambda, \tau) \in \mathbb{R}^{21\times N}$ are created in a curriculum learning scheme [6] and element-wise product is conducted on Θ_N:

$$\tilde{\Theta}_N = \Theta_N \odot \boldsymbol{M_1} \odot ... \odot \boldsymbol{M_\eta} \tag{6}$$

$\lambda \sim \mathcal{N}(5, \sigma^2), \tau \sim \mathcal{U}(1, 0.6D)$ determine the length and location of the jitter, $\eta \sim \mathcal{U}(1, l_j)$ determines the number of frames suffering the jitter. l_j is initialized as 5 and gradually increased until $\frac{N}{2}$ during training.

Formulation. With the data above, the encoder $\mathcal{E}_{trv}$ is consists of three parts ($\mathcal{F}_\epsilon, \mathcal{F}_\delta$ and $\mathcal{F}_\alpha$) can be then formulated as:

$$\hat{\Theta}_{K'} = \mathcal{E}_{trv}(\tilde{\Theta}_N) = \mathcal{F}_\epsilon(\Big[\mathcal{F}_\delta(\tilde{\Theta}_N) + \tilde{\Theta}_N)\Big] \odot \mathcal{F}_\alpha(\tilde{\Theta}_N)) \tag{7}$$

Among them, $\mathcal{F}_\delta$ and $\mathcal{F}_\alpha$ are parallel branches to regress the offset $\{\boldsymbol{\delta}_i\}_{i=1}^N$ and the attention $\{\alpha_i\}_{i=1}^N$. Each $\tilde{\boldsymbol{\theta}}_i \in \tilde{\Theta}_N$ becomes $\boldsymbol{\varphi}_i = \alpha_i(\tilde{\boldsymbol{\theta}}_i + \boldsymbol{\delta}_i)$. α_i is a scalar and $\boldsymbol{\delta}_i, \boldsymbol{\varphi}_i$ are vectors with the same shape as $\tilde{\boldsymbol{\theta}}_i$. They provide clustering features for subsequent operations.

For a composite motion $\tilde{\Theta}_N$, the number of the key pose states K are dependent on the motion content. This consensus is encouraged in two ways. In terms of architecture, a mean-shift clustering $\mathcal{F}_\epsilon$ with learnable bandwidth [49] is introduced into the last block of our network. With limited iterations, $\mathcal{F}_\epsilon$ performs clustering according to the learned feature $\boldsymbol{\varphi}_i$ among states $\tilde{\boldsymbol{\theta}}_i \in \tilde{\Theta}_N$. The sample point closest to the convergent clustering centers are regarded as the key pose states $\hat{\Theta}_{K'}$, where $K' \neq K$ is allowed. So in terms of loss design, Chamfer distance [11] $\mathcal{L}_{\text{cd}}$ is introduced to encourage the above-mentioned consensus:

$$\mathcal{L}_{\text{cd}}(\hat{\Theta}_{K'}, \Theta_K) = \sum_{\hat{\boldsymbol{\theta}} \in \hat{\Theta}_{K'}} \min_{\boldsymbol{\theta} \in \Theta_K} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2 \tag{8}$$

For each predicted key state $\hat{\boldsymbol{\theta}} \in \hat{\Theta}$, only the distance to its closest sample $\boldsymbol{\theta} \in \Theta_K$ is considered. The total loss to train the encoder in this step is:

$$\mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{cd}}(\hat{\Theta}_{K'}, \Theta_K) + w_C \mathcal{L}_{\text{C}}(\hat{\Theta}_{K'}) \tag{9}$$

where additional $\mathcal{L}_{\text{C}}$ is added to prevent the decoder from selecting the disturbed state as the key states.

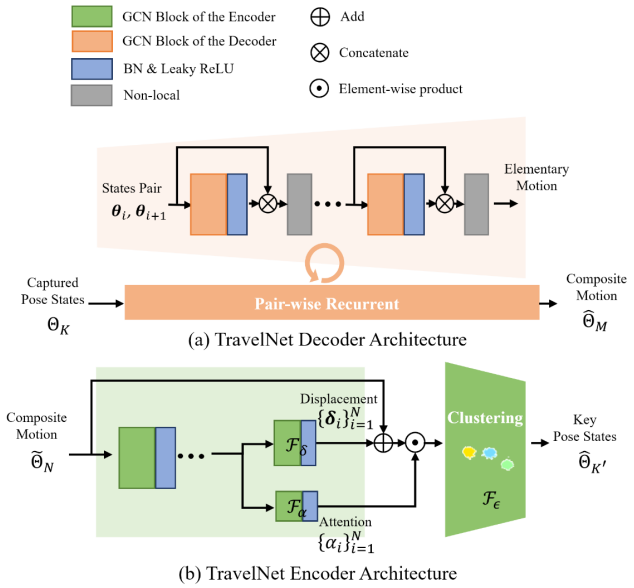


Figure 4. **TravelNet architecture.** (a) Decoder with a recurrent module; (b) Encoder with the learnable mean-shift clustering.

5.3. Encoder Fine-tuned with Captured Motions

We design the following domain adaptation strategy to fine-tune the encoder. In this step, the encoder and decoder of TravelNet are combined as the traditional auto-encoder mode to handle the real motion. First, the encoder is fed with the chronological pose states Θ_N. The key pose states extraction $\hat{\Theta}_K$ are not directly supervised but are provided to the fixed decoder to generate a motion sequence $\hat{\Theta}_M$. According to the corresponding indices of $\hat{\Theta}_K$ in the original Θ_N, generated $\hat{\Theta}_M$ is then tailored to the $\hat{\Theta}_N$ which has the same length and motion speed as Θ_N. To not only retain the details but also avoid over-fitting, a temporal smooth term is also used in this step:

$$\mathcal{L}_{\text{S3}} = \mathcal{L}_1(\hat{\Theta}_N, \Theta_N) + w_T \mathcal{L}_{\text{T}}(\hat{\Theta}_N) + w_C \mathcal{L}_{\text{C}}(\hat{\Theta}_N) \tag{10}$$

5.4. Learning Module

The detailed network architecture of TravelNet is shown in Fig. 4. Its basic learning module is the graph-based convolution block with a learnable adjacency matrix and self-attention similar to [56]:

$$f(\Theta_M; \mathbf{e}, \mathbf{w}) = \mathbf{e} \cdot (\Theta_M * \mathbf{w}) \tag{11}$$

$*\mathbf{w}$ represents a convolution layer to fuse the features at different times. $\mathbf{e} \in \mathbb{R}^{21\times 21}$ is a learnable adjacency matrix to describe the synergism among different DoFs. It is initialized as an identity matrix. The non-local attention [9, 45] is adopted for the decoder to guarantee the global consistency among the concatenated data.

For each layer of the decoder, a detailed motion is predicted according to the previous coarser one, and the pre-

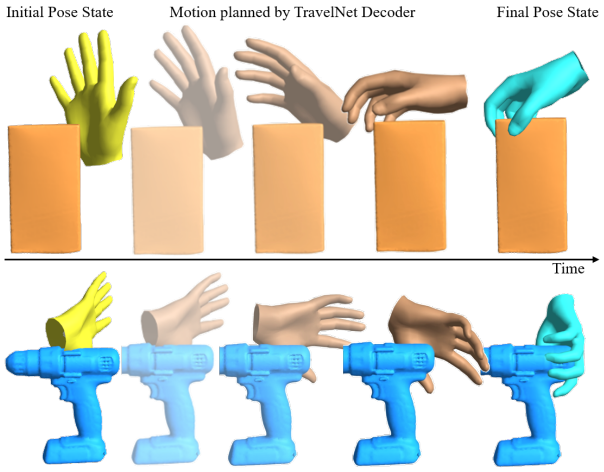


Figure 5. **Qualitative results for motion planning.** Given the start and end states, TravelNet decoder can be used in 3D space for motion planning. Please refer to the **Sup. Video** for more details.

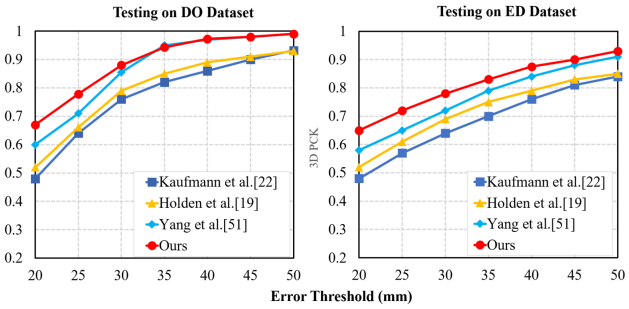


Figure 6. **Accuracy for motion reconstruction.** The left plot presents the sequential 3D PCK performance on DO dataset. The right plot presents the performance on ED dataset.

vious motion sequence is divided into two segments evenly and concatenated to the two ends of the output.

6. Experiments

6.1. Implementation Details

We use CMU-MPII [40], OneHand10K [46], Halpe [25], Frei [61], MHP [13], Hand3DStudio [57] and the first 8 sequences of STB [54] to train our image feature extraction CNN module in hand pose estimator. We adopt the network architecture in [47, 20] as the backbone of this estimator. To train the physical pose estimator, we not only use the captured data in our archive, but also some synthetic poses randomly generated in the physics engine. It consists of five layers of semantic graph convolution [56]. When selecting the initial state and final state to train the TravelNet decoder, the chronological pose states in the archive are not added as candidates, which ensures the diversity of each elementary motion. In the fine-tuning of the TravelNet encoder, we use the pose sequences in [13, 53], while the sequences in STB [54] are only used in the testing phase. We adopt Adam optimizer [23], batch normalization, and leaky-ReLu [27] for all network training. Our networks are trained on a single NVIDIA TITAN RTX GPU with a base learning rate of 1e-4. We set batch size 32 for image data, and 64 for pose sequence. The simulation frequency of the physical engine is set to 5KHz.

Method	AUC of PCK			
	DO	ED	STB	RHD
Iqbal *et al.* [20]	.672	.543	.994	-
Yang *et al.* [52]	-	-	.996	**.943**
Zhang *et al.* [55]	.825	-	.995	.901
Ge *et al.* [12]	-	-	**.998**	.920
Zhou *et al.* [58]	.948	.811	.898	.856
Ours w/o using θ	.940	.803	.890	.843
Ours w/o using γ	.947	.806	.889	.861
Ours w/o FKlayer	.950	.813	.956	.890
Ours	**.962**	**.823**	**.998**	.903

Table 2. **Accuracy for pose estimation.** Comparison with the state-of-the-art hand pose estimation methods on four public datasets as well as ablation study on our hand pose estimator.

6.2. Comparison to Related Work

Accuracy of Pose Estimation. The experimental results in Fig. 1(row 2 and row 3) show that the pose estimator becomes more accurate and plausible after using our pose representation with physical constraints. In Tab. 2, we further compare our approach to other state-of-the-art methods on test sets of RHD [60], STB [54], DO [43] and ED [33]. The following metrics are adopted to evaluate the performance: the percentage of correct 3D key points (PCK), and the area under the PCK curve (AUC) with thresholds ranging from 20mm to 50mm. Although our estimation pipeline is similar to[58], its accuracy has been greatly improved because of our physical pose representation.

Robustness of TravelNet. To explore the robustness of the method, we add perturbation to the real hand sequence, and then compare the 3D Joint Error between the reconstruction result and the original sequence. We also transfer two learning-based human body motion synthesis network [19, 22] to this task. In[22], there are only experiments to add noise to the test dataset. To verify the generalization ability for new motions, we also add new motions generated by the physics engine in this test. 3D joint error is the metric to evaluate the performances of different methods. According to the first three rows in Tab. 3, for varying degrees of perturbation and new motions, the reconstruction accuracy of TravelNet surpasses most of the existing works.

Accuracy of TravelNet. To verify the accuracy of motion reconstruction, we compare the average 3D joint error of the existing model and TravelNet on STB, DO, and ED se-

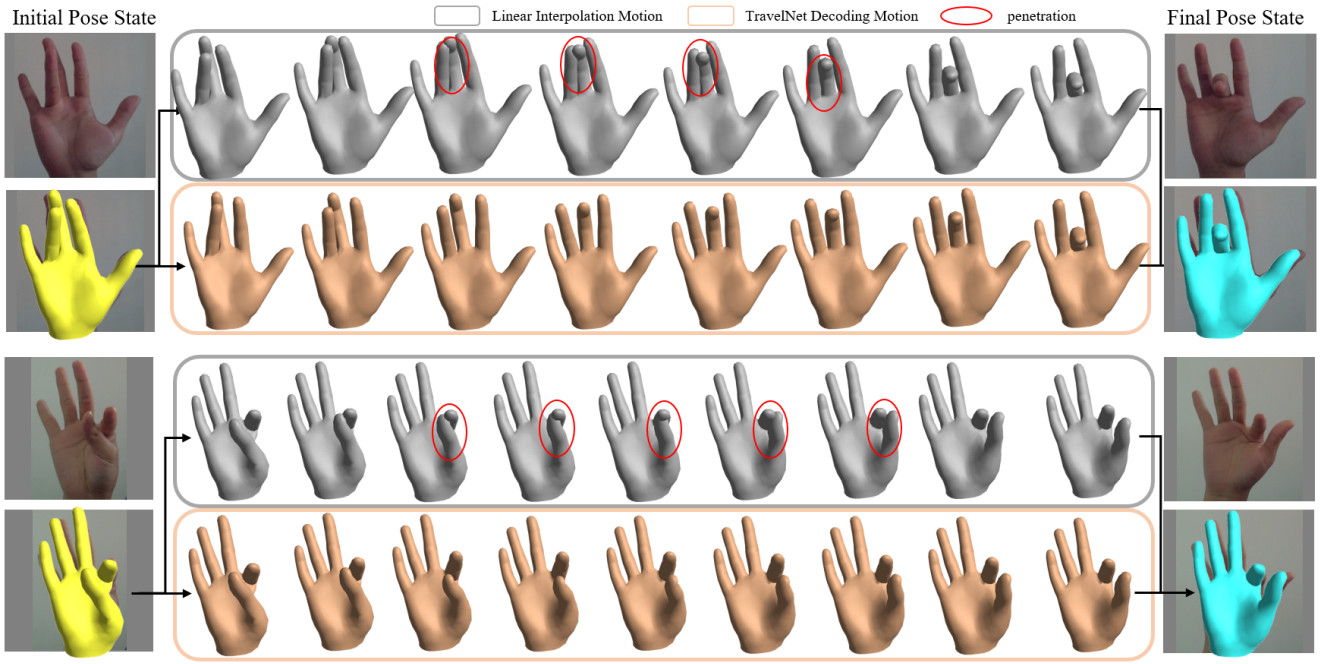


Figure 7. **Physical plausibility for in-between generations.** Given the same start and end states, the motions generated by TravelNet decoder (bronze states) contain physical constrains. While the motions generated by linear interpolation (gray states) involve penetrations (red dotted circles). Please refer to the **Sup. Video** for more details.

quence datasets. It is worth pointing out that when fine-tuning the encoder, we only use eight sequences in the STB dataset. The results of the comparison in terms of average 3D Joint Error are shown in the last 3 columns of Tab. 3, and the 3D PCKs tested on ED and DO sequential dataset are shown in Fig. 6. Compared with the frame-by-frame estimation, we find that our reconstruction accuracy has been significantly improved on a dataset such as DO that contains severe occlusions. This also shows that the TravelNet is robust to occlusions and has learned the consistency of sequential actions.

6.3. Ablation Study

Variants of Pose Estimation. We analyze the influences of using physical pose, unified bone ratio, and self-supervised term provided by FKLayer in training to our physical pose estimator in Sec. 4 and **Sup. Mat** . In Tab. 2 row 6, the necessity of physical pose $\boldsymbol{\theta}$ is verified in the following procedure. By replacing our IKNet with original IKNet in [58], the joint locations $\boldsymbol{X}$ are mapped to another pose vector $\boldsymbol{\vartheta} \in \mathbb{R}^{45}$ represented by quaternions without DoF limitations. As shown in Tab. 2 row 7 and row 8, the estimator performance will be degraded without bone ratios $\boldsymbol{\gamma}$ extraction or self-supervised term provided by FKLayer. The performance of our complete pose estimation pipeline is shown in Tab. 2 row 9. The usage of $\boldsymbol{\theta}$ has the greatest impact on the accuracy of our hand pose estimator.

Variants of TravelNet. As shown in row 4 of Tab. 3, naïve temporal smoothness optimized from the frame-wise pose estimations is first tested as a baseline. The smoothness between the two adjacent pose states is optimized, and this item is consistent with Eqn. 4. After that, we analyze the importance of clustering $\mathcal{F}_\epsilon$ and attention $\mathcal{F}_\alpha$ in the encoder, non-local of each learning block, and the collision penalty $\mathcal{L}_\mathrm{C}$ and temporal smooth term $\mathcal{L}_\mathrm{T}$ used in the training phase. The results in Tab. 3 row 5 to 9 reveal that the clustering module has the greatest impact on TravelNet; The ablation of $\mathcal{L}_\mathrm{C}$ does not greatly weaken the performance because collision has been tackled implicitly in the physical pose data generation. Although collision conflicts may be contained in $\tilde{\Theta}_N$ after random jittering, this implausible state subset will be abandoned due to the attention mechanism and clustering in the encoder; In addition, with the help from both $\mathcal{L}_\mathrm{T}$ and non-local attention, not only the piecewise smoothness but also the global smoothness is guaranteed. As shown in Fig. 5 and Fig. 7, all reconstructed motions are approximately globally smooth (refer to **Sup. Video** for more details).

Plausibility of TravelNet Decoder. Two experiments are designed to verify the plausibility of the motion generated by the decoder. Firstly, we compare it with the naïve linear interpolation by several elementary motions. As shown in Fig. 7, the generations of linear interpolation involve the penetrations of different fingers in space. By contrast, the decoder has learned these synergies and guarantees that every intermediate state on the travel path is physically plausible. We also deploy the decoder to the motion planning task with [8, 21] to determine the final state of grasping. Some qualitative results are shown in Fig. 5. Although the hand collision shape is fixed in the modeling and learning

Method	Average 3D Joint Error (mm)							
	$\sigma = 1.0$	$\sigma = 1.5$	$p = 0.2$	$p = 0.4$	Montage Motions	DO	ED	STB
Yang *et al*. [51]	28.32	31.33	27.12	31.18	20.35	18.16	18.12	9.87
Holden *et al*. for hand [19]	29.34	32.10	27.45	33.11	21.15	19.34	20.03	10.03
Kaufmann *et al*. for hand [22]	28.92	31.85	27.88	33.54	20.14	18.13	18.97	12.37
Ours w/o TravelNet.	32.57	36.14	33.73	35.72	24.31	22.57	20.75	11.06
Ours w/o $\mathcal{F}_\epsilon$	28.05	31.66	27.13	32.96	19.08	17.94	18.21	9.73
Ours w/o $\mathcal{F}_\alpha$	26.68	31.24	27.17	29.41	18.22	17.64	17.05	9.66
Ours w/o non-local & $\mathcal{L}_\mathrm{T}$	26.23	31.27	27.17	33.13	18.36	17.75	17.30	9.69
Ours w/o $\mathcal{L}_\mathrm{T}$	25.42	29.85	26.92	30.03	17.13	17.39	16.95	9.64
Ours w/o non-local	25.36	29.64	26.77	29.91	17.24	17.41	16.92	9.61
Ours w/o $\mathcal{L}_\mathrm{C}$ ✓	24.31	29.66	**26.88**	29.03	16.36	17.05	**16.70**	9.44
Ours	**24.30**	**29.65**	26.89	**29.01**	**16.35**	**17.04**	16.71	**9.42**

Table 3. **Robustness for motion reconstruction.** Comparison for robustness when adding noise, using montage motions, or using captured motions. σ indicates the average number of labeled frames disturbed by Gaussian noise. p is the masked ratio of joints in the whole sequence. The montage motion is the hand motion generated by discrete pose states from the archive using the physics engine.

Figure 8. **User controllable motion encoding.** TravelNet encoder extracts key pose states with different bandwidths on STB sequence. The key poses extracted with the learned bandwidth are in the middle line.

process, there is little self-penetration occurs even when the shape parameters change in these testing.

Plausibility of TravelNet Encoder. The learned bandwidth in the encoder is allowed to be overridden. As shown in Fig. 8, we take the counting process recorded in the STB dataset as an example to study the influence of bandwidth. We find that modifying this bandwidth affects the level-of-detail of the description accuracy of the encoder for a given motion. Although a lower bandwidth results in a more detailed motion, it is more susceptible to noise. From the results of the middle row in Fig. 8, the decoder has learned a bandwidth setting that is adaptive to the captured motion.

7. Conclusion

This paper proposes a novel paradigm to reconstruct physically plausible hand motions from monocular color images in a self-supervised manner. It is the first work to validate the physical plausibility of the hand pose and motion with the help of a physics engine only in the training phase. In our approach, physics-based DoFs are used to represent pose and animation-based key states are used to represent motion. This compactness enables TravelNet to not only reliably reconstruct but also flexibly re-edit hand motions. In the future, the representation and paradigm can be migrated to the hand-object interactions and bionic hand control retargeting control.

Acknowledgements. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1403900, National Natural Science Foundation of China (No. 61806054, 62076061), Natural Science Foundation of Jiangsu Province (No. BK20180355), Young Elite Scientist Sponsorship Program by the China Association for Science and Technology and Zhishan Young Scholar Program of Southeast University.

References

[1] Cyberglove. http://www.cyberglovesystems.com. 2

[2] Manusvrglove. https://manus-vr.com/. 2

[3] Mujoco physics engine. http://www.mujoco.org. 2, 3

[4] Vicon motion capture system. http://www.vicon.com. 2

[5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 2

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 5

[7] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 2

[8] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393. IEEE, 2019. 7

[9] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005. 5

[10] F Dincer and G Samut. Hand function: A practical guide to assessment. 2014. 3

[11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5

[12] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10833–10842, 2019. 2, 6

[13] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. Large-scale multiview 3d hand pose dataset. *Image and Vision Computing*, 81:25–33, 2019. 2, 3, 6

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[15] Genliang Guan, Zhiyong Wang, Shiyang Lu, Jeremiah Da Deng, and David Dagan Feng. Keypoint-based keyframe selection. *IEEE Transactions on circuits and systems for video technology*, 23(4):729–734, 2012. 3

[16] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. 2

[17] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019. 2

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[19] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 3, 6, 8

[20] 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. 2, 6

[21] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020. 7

[22] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927. IEEE, 2020. 3, 6, 8

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Sourabh Kulhare, Shagan Sah, Suhas Pillai, and Raymond Ptucha. Key frame extraction for salient activity recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 835–840. IEEE, 2016. 3

[25] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 382–391, 2020. 2, 3, 6

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[27] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 6

[28] Meysam Madadi, Sergio Escalera, Alex Carruesco, Carlos Andujar, Xavier Baró, and Jordi Gonzàlez. Top-down model fitting for hand pose recovery in sequences of depth images. *Image and Vision Computing*, 79:63–75, 2018. 2

[29] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *European Conference on Computer Vision*, pages 440–455. Springer, 2020. 2

[30] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 2

[31] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and

Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 2

[32] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019. 4

[33] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1284–1293, 2017. 6

[34] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4957–4965, 2016. 2

[35] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015. 2

[36] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 2

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 4

[38] Zeeshan Rasheed and Mubarak Shah. Detection and representation of scenes in videos. *IEEE transactions on Multimedia*, 7(6):1097–1105, 2005. 3

[39] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):245, 2017. 2, 3

[40] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017. 2, 3, 6

[41] Breannan Smith, Chenglei Wu, He Wen, Patrick Peluse, Yaser Sheikh, Jessica K Hodgins, and Takaaki Shiratori. Constraining dense hand surface tracking with elasticity. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 2

[42] 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. 2

[43] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*, pages 294–310. Springer, 2016. 6

[44] Ricardo VáZquez-MartíN and Antonio Bandera. Spatio-temporal feature-based keyframe detection from video shots using spectral clustering. *Pattern Recognition Letters*, 34(7):770–779, 2013. 3

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 5

[46] Yangang Wang, Cong Peng, and Yebin Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3258–3268, 2018. 2, 3, 6

[47] Yangang Wang, Baowen Zhang, and Cong Peng. Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization. *IEEE Transactions on Image Processing*, 29:2977–2986, 2019. 6

[48] Wayne Wolf. Key frame selection by motion analysis. In *1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings*, volume 2, pages 1228–1231. IEEE, 1996. 3

[49] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *ACM transactions on graphics*, 39(4), 2020. 3, 5

[50] Kazuki Yamamoto, Etsuko Ueda, Tsuyoshi Suenaga, Kentaro Takemura, Jun Takamatsu, and Tsukasa Ogasawara. Generating natural hand motion in playing a piano. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3513–3518. IEEE, 2010. 2

[51] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. In *European Conference on Computer Vision*, pages 122–139. Springer, 2020. 2, 8

[52] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2335–2343, 2019. 6

[53] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017. 2, 3, 6

[54] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016. 2, 3, 6

[55] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019. 2, 6

[56] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019. 5, 6

[57] Zhengyi Zhao, Tianyao Wang, Siyu Xia, and Yangang Wang. Hand-3d-studio: A new multi-view system for 3d hand reconstruction. In *ICASSP 2020-2020 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2478–2482. IEEE, 2020. 2, 3, 6

[58] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2020. 2, 6, 7

[59] Yueting Zhuang, Yong Rui, Thomas S Huang, and Sharad Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, volume 1, pages 866–870. IEEE, 1998. 3

[60] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017. 2, 6

[61] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 813–822, 2019. 2, 3, 6