Noname manuscript No.
(will be inserted by the editor)

Video Driven Adaptive Grasp Planning of Virtual Hand Using Deep Reinforcement Learning

Yihe Wu [1] . Zhenning Zhang [2] . Dong Qiu [1] . Weiqing Li [2] . Zhiyong Su [1, ✉]

Received: date / Accepted: date

Abstract Data-driven grasp planning can generate anthropopathic grasps, providing controllers with robust and natural responses to environmental changes or morphological discrepancies. Mocap data, which is the widely used source of motion data, can provide high-fidelity dynamic motions. However, it is challenging for non-professionals to quickly get start and collect sufficient mocap data for grasp training. Furthermore, current grasp planning approaches suffer from limited adaptive abilities, and thus cannot be applied to objects of different shapes and sizes directly. In this paper, we propose the first framework, to the best of our knowledge, for fast and easy design of grasping controller with kinematic algorithms based on monocular 3D hand pose estimation and deep reinforcement learning, leveraging abundant and flexible videos of desired grasps. Specially, we first get original grasping sequences through 3D hand pose estimation from given monocular video fragments. Then, we reconstruct the motion sequences using data smoothing based on the peek clipping filter, and further optimize them using the CMA-ES (Covariance Matrix Adaptation Evolution Strategy). Finally, we integrate the reference motion with the adaptive grasping controller through deep reinforcement learning. Quantitative and qualitative results demonstrate that our framework is able to generate natural and stable grasps easily from monocular video demonstrations, added the adaptive ability to primitive objects of different shapes and sizes in the target object library.

Keywords Virtual hand · Grasp planning · Motion generation · Deep reinforcement learning · Monocular 3D hand pose estimation

1 Introduction

Grasp planning is widely used in the field of computer animation, robotics, virtual reality, and so on. Data-driven grasp planning approaches synthesize new grasps from existing data, which can generate high-fidelity natural grasps and provide real-time responses to the perturbations of environmental changes or morphological discrepancies [29]. Mocap data is the most widely used source of motion data because of its high-fidelity dynamic information [13]. However, collecting mocap data usually requires specialized instruments such as data glove [13, 32], resulting in difficulties for non-technical users (e.g. teachers and students) to quickly get start and generate desired grasps by themselves. Furthermore, since current data-driven approaches rely on grasping datasets, their adaptive capabilities are still limited [22]. Designing an adaptive grasping controller to objects of different types and sizes, while also guaranteeing the naturalness and stability of the motion process at the same time, remains a challenging problem.

Existing data-driven grasp planning methods depend heavily on mocap data [18] or a large amount of prior information in the form of contact map [3], grasping tags [15, 22], etc. Even with enough training data, these approaches suffer from poor adaptive abilities, and thus cannot be applied to objects of different shapes and sizes directly [11]. In contrast with these prohibitive data, a more abundant and flexible source of motion data is video fragments, which can be easily obtained to describe the design goals of non-professionals [14]. However, there is just one framework that employs monocular videos, and it merely focuses on the causal dependencies among

✉ Zhiyong Su
E-mail: su@njust.edu.cn

[1] School of Automation in Nanjing University of Science and Technology, Nanjing, 210094, China
[2] School of Computer Science and Engineer in Nanjing University of Science and Technology, Nanjing, 210094, China

hand interactions [12]. Thus, the proposed model can only generate schematic grasps with inferior naturalness. With the development of video-based hand pose estimation technologies, it is fairly straightforward to make full use of video resources to learn customized adaptive grasps. Nevertheless, existing video-based hand pose estimation approaches typically give frame-wise processes with poor spatio-temporal consistency [35]. Besides, motion noise always exists due to the factors such as occlusion or low video resolution [6]. Thus, the quality of estimation results still falls behind the elaborately collected mocap data.

In this paper, to the best of our knowledge, we propose the first framework to learn an adaptive grasping controller from a single monocular video fragment without the complex force closure solution process. It integrates 3D hand pose estimation and deep reinforcement learning, and focuses on the kinematics algorithm during grasping and motion synthesis. In order to customize the desired grasping sequences from monocular videos, firstly we employ the monocular hand pose estimation framework [35] to get the original grasping sequences. Then, we obtain the reference motion by reconstructing the sequences and adapting them with the physics-based hand model. To develop an adaptive grasping controller, we employ a deep reinforcement learning algorithm combined with the reference motion. Through the offline training, the simulated hand is able to imitate the reference motion while observing the environmental states, providing natural and stable grasps, eventually.

The contribution of this paper can be summarized as follows:

1) A monocular video-driven method for customizing smooth grasping reference motion from given RGB video fragments. For non-technical users, they can freely record required videos from accessible monocular cameras. Thus, the reference motion can be easily produced without prohibitive infrastructures.
2) A deep reinforcement learning-based framework for learning an adaptive grasping controller. While ensuring the naturalness and stability of generated grasps, our controller can grasp target objects of different shapes and sizes with kinematic algorithms. Our framework can also perform well in the case of different hand morphologies.

We review the related works and analyze the current problems in Section 2. Then we propose the main framework of this paper in Section 3. Section 4 and Section 5 introduce the algorithms of video-based reference motion generation and adaptive grasp planning based on deep reinforcement learning, respectively. Furthermore, their results are discussed in Section 6. Finally, we evaluate our method and draw the conclusion in Section 7.

2 Related Work

2.1 Data-driven grasp planning

Grasp planning has a long history in the field of computer animation, virtual reality and robotics [9]. Among them, data-driven grasp planning approaches typically synthesize new grasps from existing data, remaining the research hotspot in the field of grasp planning [15]. Wang et al. segmented complex objects into object elements based on their superquadrics model, and then further proposed the optimal grasps using simulated annealing algorithms combined with mocap grasping data [29]. Starke et al. trained the autoencoder by mocap data, which is able to generate new grasps [27]. Kopicki et al. proposed a few-shot algorithm to demonstrate the grasp trajectory recorded by a depth camera [18]. Liu et al. analyzed the grasps corresponding to different target objects as the dataset with labels [22]. Combining the consistency loss function and collision loss function, they finally established the grasping controller using deep neural network. Brahmbhatt et al. trained the grasping controller that is stable and functional for the objects [3]. However, it required massive contact maps. Wang et al. proposed a generative model for human-object interactions from videos [12]. Nevertheless, the generated hand motion was supported by splines, which is interpolated between different hand pose stored with the object.

To summarize, existing data-driven approaches [3, 12, 18, 22, 27, 29] are robust for slightly environmental perturbations, but they generally require mocap data [13, 29] or a large amount of prior information[3, 18, 22, 27]. Although few approaches can generate schematic grasps from monocular videos, the naturalness of the grasps is relatively poor [12].

2.2 Monocular 3D hand pose estimation

In the field of technical interactive exhibitions, there are some attempts to design systems using monocular cameras[25]. What's more, in the field of AR and VR, high-quality hand interaction can increase user immersion and improve teaching effectiveness[4]. For this reason, more and more attention has been paid to research of hand pose estimation recently. Monocular 3D hand pose estimation approaches usually employ supervised learning algorithms, combined with feature extracting and inverse kinematics techniques. Mueller et al. proposed the CycleGAN framework, which predicts the position of joints through the given monocular videos [10]. However, since they ignored the joint rotation, it is difficult to generate robust grasping estimation. Zhang et al. developed

the HAMR framework, which is an end-to-end method to obtain mesh parameters of the deformable hand from the regression model [30]. It can get smooth motion sequences and address the problem of occlusion. Nevertheless, the accuracy of the pose estimation is relatively low because it is trained with a weakly supervised learning algorithm. Zhou et al. made full use of multi-modal grasping data [35], solving the difficulty of insufficient data set. But, the spatio-temporal consistency of the estimation is poor, which may cause jitter or joint accuracy error.

In conclusion, since existing monocular 3D hand pose estimation methods are frame-wise approaches [10, 30, 35], they can not be used as templates directly because of their low spatio-temporal consistency and motion noise.

2.3 Physics-based characters from monocular videos

Recently, we have witnessed an overwhelming growth of new approaches to deal with the tasks of controlling physics-based characters. Some state-of-the-art approaches combined reinforcement learning algorithms with mocap data, or further produced the mocap data from monocular videos. Based on DeepMimic [32], Peng et al. directly obtained reference motions from monocular videos through OpenPose [5] , as well as 3D human body mesh recovery framework HMR [17]. Their work was able to generate highly dynamic human skills from video datasets of YouTube [31]. Similarly, Lee et al. developed a controller of the figure skating using DQN, combined with the reference motion estimated from videos [34]. Recently, Shimada et.al presented PhysCap [26]. It is the first framework for physically-plausible, real-time and marker-less human 3D motion capture from a single monocular camera.

Although the research of character motion has gradually matured, the interactions between virtual hand and target objects are less explored [2]. And, it remains a challenging work in the field of character animation.

3 Overview

Our framework covers two main steps to learn the adaptive grasping controller from given video fragments: video-based reference motion generation stage, and adaptive grasp planning stage, as shown in Fig. 1.

In the video-based reference motion generation stage, we first extract frame-wise pose from monocular video fragments using the monocular 3D hand pose estimation technique [35]. Then, we employ data smoothing and trajectory optimization to reconstruct the motion trajectory [20]. Finally, we bind the sequence to the physical-based anthropomorphic hand model. Thus, a reference motion trajectory is optimized, which is used as the imitation template for the grasping controller.

In the adaptive grasp planning stage, the adaptive grasping controller is trained with deep reinforcement learning in the training phase, added the target objects according to the grasping tasks. In the inference phase, with the input target objects for inferencing, our controller trained in the training phase is able to generate real-time grasps.

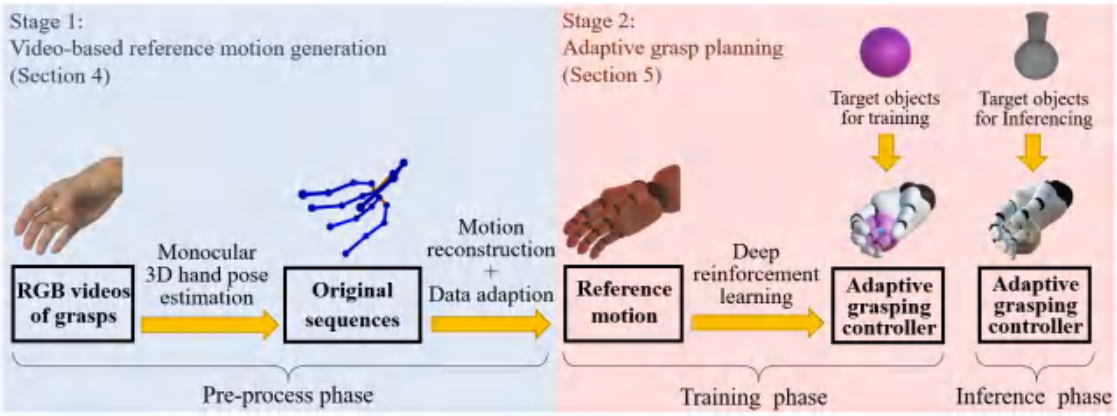


Fig. 1 Pipeline of our system.

4 Video-based reference motion generation

In this section, we first produce the original grasping sequence by 3D hand pose estimation technique. Then we reconstruct the motion trajectory through data smoothing and trajectory optimization process. Finally, we bind the data to the physically-plausible anthropopathic hand model.

4.1 Original grasping sequence generation

The original rotation sequence of each hand joint is extracted by Minimal-Hand [35], which is a monocular 3D hand pose estimation framework for predicting the hand pose in each frame of given video fragments. At the beginning, heat map data is obtained from the input RGB images through the 2D feature extraction. Then, a 3D detector is used to obtain the position and rotation data about the joints. Finally, the original hand motion sequence is achieved by inverse kinematic algorithms.

Since poses are predicted independently for each frame, simply imitating the trajectory tends to produce motions that exhibit artifacts due to inconsistent predictions across adjacent frames. These artifacts can hinder the simulated character's ability to reproduce intended motions.

4.2 Motion reconstruction

The motion reconstruction stage reconstructs a new kinematic trajectory [31], which can reconcile the individual predictions, as well as mitigate artifacts. After the motion reconstruction, the original motion sequence is more amenable for imitation. At this stage, we first process the original sequence using data smoothing based on peek clipping filter. Then, we employ CMA-ES for further trajectory optimization.

4.2.1 Data smoothing based on peek clipping filter

The peak clipping filter is employed to enhance the spatio-temporal consistency of the predictions across adjacent frames. The filter compares the angles estimated joint by joint and frame by frame:

$$\Delta r_t^i = r_t^i - r_{t-1}^i, \tag{1}$$

where r_t^i denotes the angle of the i-th joint at timestep t, and Δr_t^i denotes their angle deviation. The peak clipping threshold γ ($\gamma \geq 0$) is set manually according to the speed of finger joints from video fragments. It is used to limit the angle deviation as:

$$\Delta r_t^i = \begin{cases} \gamma, & \text{if } \Delta r_t^i \geq \gamma \\ \Delta r_t^i, & \text{if } \left|\Delta r_t^i\right| \leq \gamma \\ -\gamma, & \text{if } \Delta r_t^i \leq -\gamma \end{cases}. \tag{2}$$

In Eq.2, the absolute angular deviation of the joint i at timestep t and t-1 satisfies $\left|\Delta r_t^i\right| \leq \gamma$, while the excess angle will be limited, as shown in Fig.2. The operation is repeated until all joints and all frames are processed.

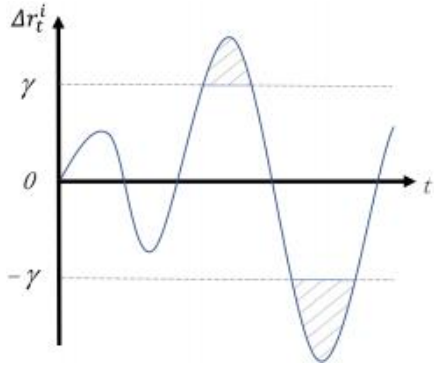


Fig. 2 Schematic diagram of peek clipping filter.

4.2.2 Trajectory optimization based on CMA-ES

Trajectory optimization [19] is employed to find a set of corrective offsets to further denoise and smooth the trajectory processed by section 4.2.1. Our system employs CMA-ES to address the optimization problem. CMA-ES is a sampling-based method that iteratively evaluates a number of random samples and updates the sample distribution according to their costs. For each CMA-ES sample, our system applies the corrective offsets $\boldsymbol{x}$ to the corresponding motion fragments and simulates two optimization windows at the same time. The objective function is defined as:

$$\min \begin{cases} E_1 = \sum_t \|(\boldsymbol{r}_t + \boldsymbol{x}_t) - (\boldsymbol{r}_{t-1} + \boldsymbol{x}_{t-1})\| \\ E_2 = \sum_t \|\boldsymbol{x}_t\| \end{cases}, \text{ s.t. } -\chi \leq x_t^i \leq \chi. \tag{3}$$

To smooth the rotation of the hand joints, Eq.3 is used to search for the minimum compensation value between adjacent frames, where $\boldsymbol{r}_t$ denotes the rotation vector of the 20 joints at timestep $t(t \in c_i)$, x_t denotes the corresponding compensation value vector at timestep t, and $\chi(0 \leq \chi)$ denotes the threshold of the corrective offsets. In this way, we transform the motion trajectory denoising problem into an optimization problem of a $20 \times n$ compensation value matrix over the object of Eq.3.

For a long and complex reference motion sequence consisting of many control fragments, the optimization variable set $\boldsymbol{X}$ contains a large number of parameters, making the optimization prohibitively expensive and prone to fall into local minima. To mitigate this problem, we employ a sliding window scheme [20], which is illustrated in Fig.3.

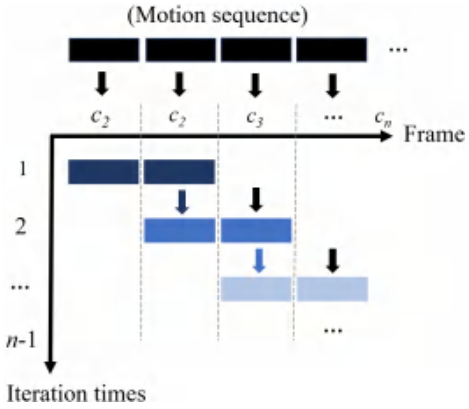


Fig. 3 Schematic diagram of sliding window.

Specifically, our system evenly divides an entire sequence into optimization windows c_i, and then optimizes two consecutive windows at each iteration. The first two windows are optimized together with all the corrective offsets initialized to zero. When the optimization finishes, the first window is removed from the optimization problem, and the second as well as the third windows are optimized together. The optimized corrective offsets for the second window are kept as the initial solution for this new optimization problem, while the corrective offsets for the third window are initialized to zero. This process is repeated until the entire motion sequence is optimized. Finally, we obtain the final smooth trajectory.

4.3 Data adaptation

In this section, we design a physical-based multi-fingered hand model. To meet the need of interacting with other objects, the agent (represented as a white hand) should strictly follow the laws of physics including properties of mass, collider, as well as hinge joints [21]. Mass defines the precise mass of each finger joint and interactive objects, so that they can be influenced by gravity or Newton's law of motion. Collider provides collision volume for rigid bodies, which promotes the interaction between objects. Hinge joint is utilized to link each finger joints. It also provides corresponding torque and damping [8].

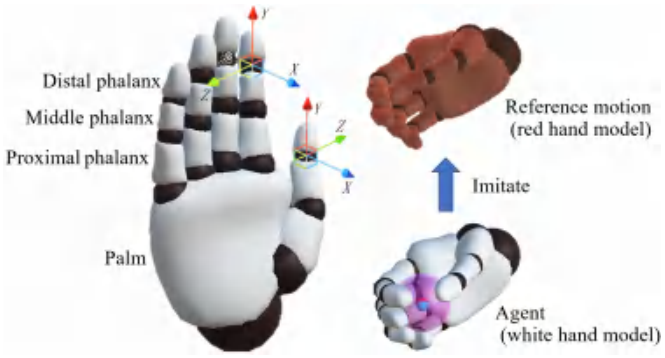
We define the hierarchical hand model according to the anatomical structure of the real human hand [33], which contains 15 finger joints, 1 palm, and 20 DOFs (Degree of Freedom). The connection of joints and coordinate system is illustrated in Fig.4(a). The mass of each finger joint is defined as 0.03kg, and the palm is 0.5kg. In order to perceive interaction and avoid penetration, spherical collision bodies are applied to each fingertip.

Table 1 Joint angle constrains of physical hands.

Joint types	X-axis rotation()	Y-axis rotation ()	Z-axis rotation ()
Proximal phalanx(thumb)	-30∼60	locked	-50∼50
Middle phalanx(thumb)	0∼90	locked	locked
Distal phalanx(thumb)	0∼90	locked	locked
Proximal phalanx(others)	-5∼90	locked	-20∼20
Middle phalanx(others)	0∼90	locked	locked
Distal phalanx(others)	0∼45	locked	locked

We abstract the structure composed of joints as a rod-like hinge structure. The joint constraints are shown in Table 1.

After the establishment of the physical hand model, we also bind the reconstructed motion sequence to the physical hand (colored red) in the Unity3D environment, so that the reference motion is finally encapsulated. Note that the red hand merely serves as the template for the agent (white hand) to imitate in the training phase, as depicted in Fig.4(b).



a) Hierarchical model b) Red hand and white hand

Fig. 4 Physical hand model.

5 Adaptive grasp planning based on deep reinforcement learning

In this section, we first define the target-grasping tasks as an optimization problem. Then, we describe the network structure of the grasping controller. Finally, we model the target-grasping task as the training goal of reinforcement learning.

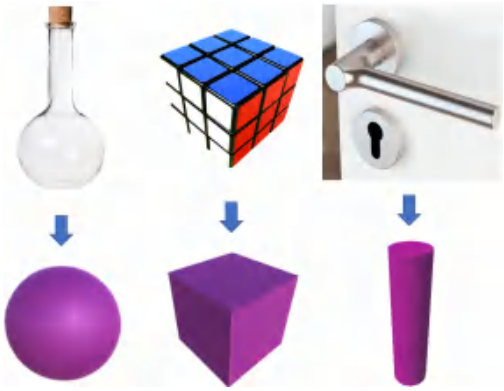
5.1 Task

We define the agent as the simulated white hand in the reinforcement learning task. According to the classification theory in [29], the primary part of the interactive objects can be abstracted into multiple basic objects such as spheres, cubes, and columns. Besides, every grasping motion has a target object in these elements. We take three basic objects as examples and then add them to the target object library, as shown in Fig.5.

The inference phase refers to the stage where the off-line model works in real-time. In the inference phase, we define the task as grasping objects of specific shapes with arbitrary sizes in the target object library. Meanwhile, the generated grasps are required to be natural and stable. A complete process for the agent to accomplish a series of tasks is known as an episode [36]. In each timestep of every episode, the agent (white hand) observes the target object and the kinematic information given by the reference motion (bind to the red hand). Then, the agent updates its state, and adjusts itself to match the object depending on the off-line policy. Finally, it obtains feedback as rewards. The complete grasping motion sequence can be generated after repeating the observation-interaction-feedback steps until the end of this episode.

5.2 Network

Each policy in the deep reinforcement learning is represented by a neural network that maps the given state s and the goal g to a distribution over the action $\pi(a \mid s, g)$. The inputs are processed by a fully-connected layer



a) Sphere b) Cube c) Cylinder

Fig. 5 Abstraction of the target objects.

with 199 units, which represent the observation of states from agents. Meanwhile, a linear output layer with 20 units represents the quantified actions of the agent. The structure of our network is shown in Fig.6. Moreover, sigmoid activations are used for all hidden units. Details of states and actions will be discussed in the following section.

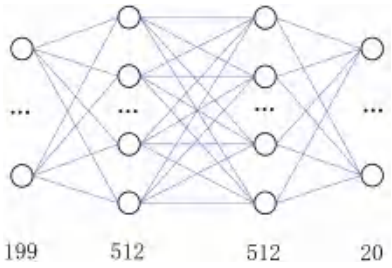


Fig. 6 The structure of our neural network.

5.3 States

In our task, the state s refers to the observation of the external environment perceived by the agent. In the neural network shown in Fig.6, the states represented by the input vector are represented as a 199 DOFs vector.

$$\boldsymbol{s}_t = \{\boldsymbol{p}_t, \boldsymbol{r}_t, \boldsymbol{v}_t, \boldsymbol{\omega}_t, \phi_t, \lambda, d_t^{\rho}, d_t^{c}\}, \tag{4}$$

where $\boldsymbol{p}_t$, $\boldsymbol{r}_t$, $\boldsymbol{v}_t$, and $\boldsymbol{\omega}_t$ describe the configuration of the hand with features consisting of the positions of each link relative to the palm, their rotations expressed in quaternions, and their linear and angular velocities, respectively.

Since the target pose from the reference motions varies with time, the phase variable ϕ_t is also included in the state vector. In this representation, $\phi_t = 0$ denotes the start of a motion, and $\phi_t = 1$ denotes the end. Besides, λ represents the size factor of the target object observed by the agent,

$$d_t^{\rho} = \sum_i \|\Delta\rho\|_t^i, \tag{5}$$

where d_t^{ρ} denotes the sum of the Euclidean distance between the surfaces of fingertips and the target object at timestep t:

$$d_t^{c} = \|\Delta c\|_t, \tag{6}$$

where d_t^{c} represents the Euclidean distance deviation between the centroid of the fingers and the centroid of the object at timestep t.

5.4 Actions

The action a is the quantitative value of the agent's behavior based on the states and the policy learned in the training phase. Since the reference motion in Section 4 only provides kinematic information in the form of target pose, the policy is also responsible for determining which actions should be applied at each timestep in order to grasp the target object firmly. The output of the neural network in Fig.6 is represented as $\boldsymbol{v}_t$, which refers to the 20 DOFs velocity vector of each joint around the X-axis or Z-axis at timestep t.

5.5 Rewards

We formulate the tasks as reward functions in reinforcement learning algorithms. Inspired by DeepMimic [32], our reward R_t at each timestep t consists of two terms, encouraging the simulated hand to imitate the reference motion while also grasp the target object firmly:

$$R_t = w^I R_t^I + w^G R_t^G, \tag{7}$$

where R_t^I and R_t^G represent the imitation reward and adaptive penalty, respectively, w^I and w^G are their respective weights. Inspired by [1], to alleviate the problem of excessive initial hand shape penalty, w^G is given by:

$$w^G = \begin{cases} \frac{\phi_t}{\phi_{\text{trans}}}, 0 \le \phi_t \le \phi_{\text{trans}} \\ 1, \quad \phi_{\text{trans}} \le \phi_t \le 1 \end{cases}, \tag{8}$$

where ϕ_{trans} denotes the transition phase.

5.5.1 Imitation rewards

Based on reverse reinforcement learning [23], the imitation reward is designed to incentivize the simulated hand to match the reference motion frame by frame. Compared with simply completing the task, the reward makes it easy to avoid the self-contact of the fingers, and further makes the generated grasp more natural.

The reward with three weights w^r, w^v and w^e is defined as follows:

$$R_t^I = w^r r_t^r + w^v r_t^v + w^e r_t^e, \tag{9}$$

where

$$r_t^r = e^{-\sum_i \|\Delta \boldsymbol{p}\|_t^i}, \tag{10}$$

$$r_t^v = e^{-\sum_i \|\Delta \boldsymbol{v}\|_t^i}. \tag{11}$$

The rotation reward r_t^r and the velocity reward r_t^v encourage the virtual hand to match the joint orientation of the reference motion at each timestep t. Specially, $\boldsymbol{\Delta p}$ and $\boldsymbol{\Delta v}$ represent the orientation and velocity deviation between the simulated hand and the reference motion, respectively. It is computed as the deviation between the joint orientation quaternions of the simulated hand and those of the reference motion.

Similarly, the end-effector reward r_t^e encourages the fingertips to match the positions from the reference motion at each timestep t. Since the human hand is a hierarchical structure with the parent-child relationship, the closer to the fingertips, the more easily the positions of the finger joints will deviate from the reference motion. The end-effector reward is defined as:

$$r_t^e = e^{-\sum_i |\Delta \boldsymbol{\rho}|_t^i}, \tag{12}$$

where $\boldsymbol{\Delta \rho}$ specifies the position deviation between the fingertips and those from the reference motion at timestep t, with $\boldsymbol{\rho} \in \boldsymbol{p}_t$.

5.5.2 Adaptive penalties

In order to stably grasp target objects of different shapes and sizes, the adaptive penalty is involved by penalizing the agent for failing to meet task-specific objectives. Note that stiff imitation of reference motion sequences may generate natural grasps, but they can hardly grasp any target stably. We define the adaptive penalty as:

$$R_t^G = w^\rho r_t^\rho + w^n r_t^n + w^c r_t^c, \tag{13}$$

where w^ρ, w^n and w^c are weight coefficients.

The end-effector distance penalty r_t^ρ [7] is computed from the distance between the fingertips and the target object at timestep t. It incentivizes the fingertips to contact the target object as soon as possible instead of stiffly imitating the reference motion. For the fingertip i of each finger, we formulate:

$$r_t^\rho = \begin{cases} -\frac{\|\Delta\boldsymbol{\rho}\|_t^i}{\max(\|\Delta\boldsymbol{\rho}\|)}, & \text{if the fingertip contacts} \\ 0, & \text{if the fingertip does not contact} \end{cases}. \tag{14}$$

r_t^n denotes the non-contact penalty of the fingertips at timestep t. In order to ensure contact between the fingertips and the object surface, for each fingertip, the penalty is given by:

$$r_t^n = \begin{cases} 0, & \text{if } \phi_t \leq 0.6 \\ -0.2, & \text{if } 0.6 \leq \phi_t \text{ and no fingertip contacts} \end{cases}. \tag{15}$$

r_t^c is the centroid penalty between the fingertips and the target object at timestep t. The presence of end-effector distance penalty and non-contact penalty merely solve the problem of whether the object is in contact, but the distribution of touch points on the object surface still needs to be optimized. Inspired by [24], we believe that the shorter distance between the object's center of mass and the centroid of the target object, the less effect of inertial and gravitational forces will be on the grasps. In other words, the grasps are more stable. The centroid penalty is given by:

$$r_t^c = -\frac{\|\Delta\boldsymbol{c}\|_t}{\max(\|\Delta\boldsymbol{c}\|)}. \tag{16}$$

We formulate the centroid of the fingers' position at timestep t as:

$$\boldsymbol{c}_t = \frac{\sum_i \left(m^i \boldsymbol{\rho}_t^i\right)}{\sum_i m^i}, \tag{17}$$

where m^i represents the mass of the i-th fingertip.

6 Experiments and results

In this section, we present the results of reference motion and adaptive grasping controller, respectively. Note that, users are more concerned about animation effects rather than physical properties in some scenarios, such as augmented reality (AR) and virtual reality (VR). To make the entire behavior generation process more lightweight, in this paper, we focus on the kinematics algorithm and the motion quality in the grasping process. And, we omit the complex force closure solution when fingers interact with target objects. Therefore, it is difficult to conduct direct performance comparison between our method and existing grasping methods based on robotic arms [3, 18] or dynamics [9, 21, 27].

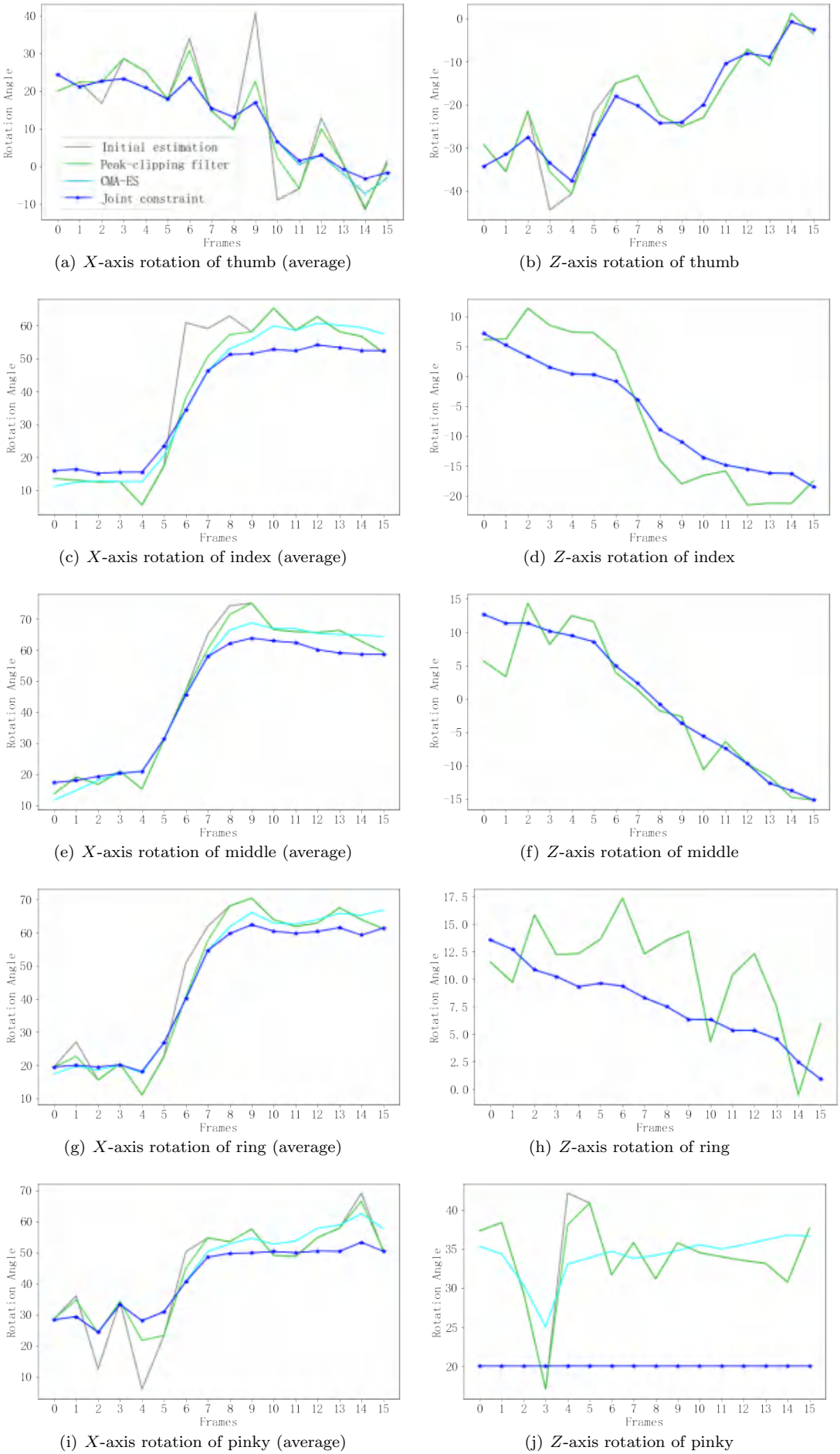
In terms of hardware, our experiment involves a server equipped with an Intel i7-9700KF processor, a NVIDIA GTX 2070 graphics card, as well as a 16GB RAM. In terms of software, the task environment is built based on Unity3D game engine with ML-Agents [16] plug-in.

6.1 Generation of the reference motion

6.1.1 Motion reconstruction

We take the motion of "cylinder-grasping" sequence (Fig.8(a)) as an example, and the initial trajectories from Minimal-Hand are shown as the gray lines in Fig.7. In the filtering stage, we set the peak clipping thresholds $\gamma_{tx} = 22$, $\gamma_{tz} = 14$, $\gamma_{ox} = 23$, $\gamma_{oz} = 21$. As is depicted in Eq.2, they represent the angle threshold of thumb's X-axis, thumb's Z-axis, other fingers' X-axis, other fingers' Z-axis, respectively. The filtering results are shown as green lines. In the trajectory optimization stage of Eq.3, we divide the processed sequence into 16 windows, and then set the step parameter σ=0.3 and the compensation threshold χ =7 for the CMA-ES optimization. The results are reported as the cyan lines. The reference motion bound to the virtual hand (red) needs to meet the joint constraints defined in Section 4.3, and the final results are given by the blue lines in Fig.7 with stars.

"Average" in the graph refers to the average Euler rotation of the three DOFs for each finger's joints. According to the results, angle noise in original trajectories can be effectively suppressed, making the final reference motion more amenable for imitation. In Fig.7(j), the original estimation of the pinky's Z-axis exceeds the bodily constraints. It may be caused by the occlusion of other fingers in the video, resulting in the error estimated by Minimal-Hand.



(a) X-axis rotation of thumb (average)

(b) Z-axis rotation of thumb

(c) X-axis rotation of index (average)

(d) Z-axis rotation of index

(e) X-axis rotation of middle (average)

(f) Z-axis rotation of middle

(g) X-axis rotation of ring (average)

(h) Z-axis rotation of ring

(i) X-axis rotation of pinky (average)

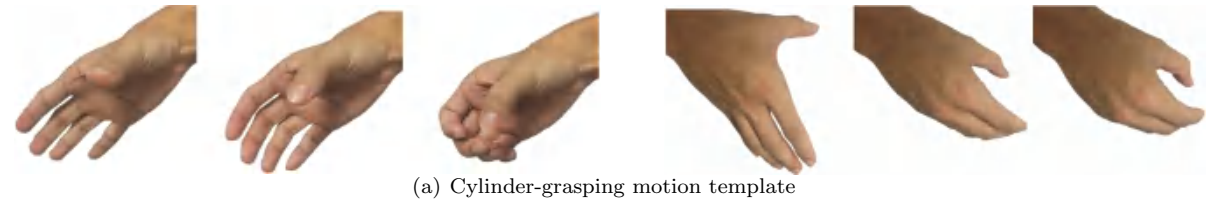
(j) Z-axis rotation of pinky

Fig. 7 Joint angles of fingers over time.

6.1.2 Data adaption

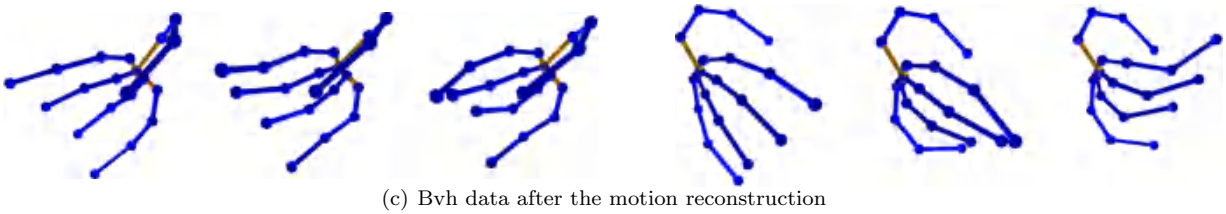
The template of "cylinder-grasping" motion is demonstrated in Fig.8(a). The spatial position of each finger joint that changes over time composes the trajectories of the motion sequences. Besides, the initial estimation from Minimal-Hand is shown in Fig.8(b)). After the motion reconstruction stage, the bvh data of human right hand are shown in Fig.8(c). The final results of the reference motion bound to the virtual hand (red) are demonstrated in Fig.8(d).

The qualitative results show that the original estimation trajectory from Minimal-Hand may produce unnatural pose with joint misalignment or jitter, as shown in Fig.8(b). They can be effectively alleviated in the motion reconstruction stage, and will eventually disappear in the reference motion.

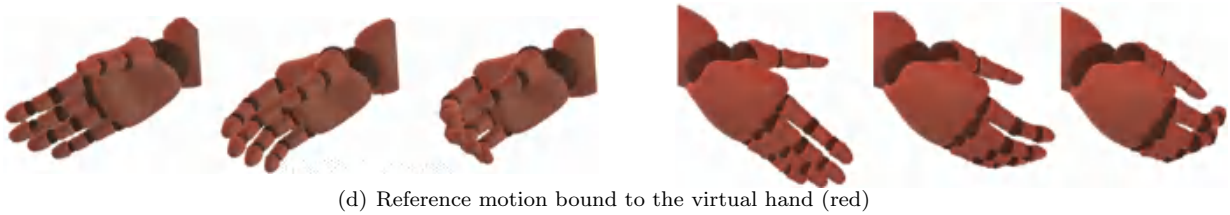


(a) Cylinder-grasping motion template

(b) Estimation from Minimal-Hand



(c) Bvh data after the motion reconstruction



(d) Reference motion bound to the virtual hand (red)

Fig. 8 Adaption of reference motion.

6.2 Generation of adaptive grasps

We analyze the qualitatively and quantitatively results from three different experiments: ablation, adaption and retarget. In the training phase, we set the time scale of 10 and parallel training of 64 agents, combined with CUDA 10.0 and cuDNN7.4 for GPU-based acceleration. We define the reward weights w^r=0.4, w^v=0.2, w^e=0.4, w^ρ=0.25, w^n=0.25, w^c=0.5, and the phase variable ϕ_{trans} =0.4.

6.2.1 Ablation

To evaluate the impact of our designed controller, we compare our full policy against alternative training schemes that disable some of the components. Comparisons include imitation-only policy and adaptation-only policy, as well as DeepMimic policy and imitation-adaptation policy.

Fig.9 compares the learning curves with different policies. "Imitation-only" policy is the policy with the imitation weight w^I of 1 and the adaptive weight w^G of 0 in Eq.7. Since no penalty is involved in this case, the

initial value of the reward is 0, and the curve has a limited upside. "Adaption-only" policy denotes the situation where the imitation weight w^I is 0 and w^G is 1. Since there are just penalties in this case, the theoretical maximum reward is 0. "DeepMimic" policy refers to the policy of [32], with w^I of 1 and w^G of 1. Finally, our "Imitation+Adaptation" policy refers to the case where w^I is 1, and w^G satisfies Eq.8. Compared with the "DeepMimic" policy, our policy has faster convergence rate. Since the dynamic weight in Eq.7 mitigates the effect of excessive adaptive penalty at the start of each episode, our strategy is more likely to avoid local optimal solutions.

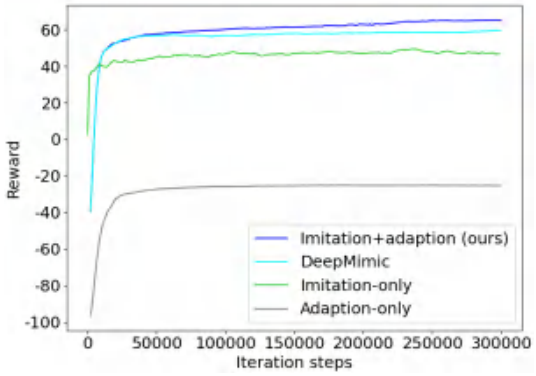


Fig. 9 Learning curves of policies in the training phase.

Snapshots of grasps with three policies are available in Fig.10. The little blue sphere in the picture represents the centroid of the fingertips, and the little red sphere represents the center of mass of the target object. Since the grasps of "DeepMimic" policy and our "Imitation+Adaptive" policy are little distinguishable, the analysis of the "DeepMimic" policy is omitted in the following part. Table 2 also summarizes the performance of the policies.

(a) "Imitation-only" policy

(b) "Adaption-only" policy

(c) "Imitation + adaption" policy

Fig. 10 Grasps with different policies.

We involve sequence similarity [28] between the generated grasp motion and the reference motion to quantize the naturalness of the results:

$$\mathrm{Sim}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{l}{\max(n, m)}, \tag{18}$$

Table 2 Quantitative results of the ablation experiment.

Policy name	Sim($\boldsymbol{X}$, $\boldsymbol{Y}$)	η
"Imitation-only" policy	88.01%	8.53%
"Adaption-only" policy	74.18%	1.29%
"Imitation+adaption" policy	**85.97%**	**1.52%**

where $\boldsymbol{X} = \{x_1, x_2, ..., x_n\}$ denotes the grasping sequence produced by the simulated hand (white), $\boldsymbol{Y} = \{y_1, y_2, ..., y_n\}$ denotes the reference motion sequence from the video (red), n and m denote the respective lengths of these sequences, respectively, and l represents the common subsequence length. We define d_{min}=0.5 as the threshold, then the common subsequence specifies the sequence whose least square distance in the sequence $\boldsymbol{X}$ and $\boldsymbol{Y}$ are less than d_{min}. The relative value of the centroid deviation is used to quantize the stability of the generated grasps. We formulate the centroid deviation ratio $\boldsymbol{\eta}$ as:

$$\boldsymbol{\eta} = \frac{\|\boldsymbol{\Delta c}\|_{t'}}{\lambda}, \tag{19}$$

where t' refers to the timestep when all the fingers touch the object surface at the same time. The results are the average value of 10 consecutive episodes. In the following experiments, we will continue to employ the quantitative methods above.

The agent will struggle to synchronize the reference motion frame by frame in the case of the "Imitating-only" policy. Qualitatively, the motions generated by the policy closely resemble the reference fragments, but the centroid deviation ratio which quantizes grasping stability is inferior, and fingertips often fail to touch the object surface, as shown in Fig.10(a). The policy of "Adaption-only" appears to grasp more firmly, but the agent never learns to act like a real human hand, with artifacts such as self-contact of the fingers, as shown in Fig.10(b). Our "Imitation + Adaptive" policy is proven to be the best control policy of the simulated hand, and the naturalness and stability quantitative values are relatively high, as shown in Fig.10(c).

6.2.2 Adaptive abilities

In order to evaluate the adaptive ability of our policy, we first choose different objects in the target object library to test the shape adaptive ability of our policy. Then we change the size factor λ to test the size adaptive ability of our policy.

6.2.2.1 Shape adaptive ability

In this section, the trained policy of grasping spheres is employed in the tasks of grasping cubes, cylinders and spheres, respectively. Besides, the adaptive ability to grasp objects of different shapes is discussed. The grasps are shown in Fig.11, and the performance statistics are shown in Table 3.

Table 3 Quantitative results of the shape adaption task.

Task name	Sim($\boldsymbol{X}$, $\boldsymbol{Y}$)	η
Cube-grasping task	83.89%	1.74%
Cylinder-grasping task	82.91%	1.72%
Sphere-grasping task	86.24%	1.52%

The results show that our policy can generate natural and stable grasps for target objects of different shapes in the target object library. Meanwhile, good adaptive ability makes our policy insensitive to hyper-parameters. Thus, the defined hyper-parameters can be well generalized to multiple motions.

6.2.2.2 Size adaptive ability

In this section, we change the size factor of the target object to analyze the adaptive ability for grasping objects of specific shapes and different sizes. Specially, the size factors λ are slightly outside the range of those in the training phase: in the training phase, the size factor of the training object is randomly sampled of $\lambda \in [0.5, 1]$; in the inference phase, we manually set the size factors to 0.3, 1.2 and 0.75, respectively.

The grasps for the different sizes of spheres are shown in Fig.12, and the quantitative results are available in Table 4.

The results indicate that our policy is able to generate stable grasps for objects of moderate size (λ =0.75), as well as larger size (λ =1.2) and smaller size (λ =0.3). Since the grasps generated in extreme cases are quite different from the initial reference motion, the Sim($\boldsymbol{X}, \boldsymbol{Y}$) is relatively low and the $\boldsymbol{\eta}$ is high.

(a) Cube-grasping task

(b) Cylinder -grasping task

(c) Sphere-grasping task

Fig. 11 Grasps of target objects of different shapes.

(a) Size factors λ= 0.3

(b) Size factors λ= 1.2

(c) Size factors λ= 0.75

Fig. 12 Grasps of target objects of different sizes.

Table 4 Quantitative results of the size adaption task.

λ	Sim($\boldsymbol{X}$, $\boldsymbol{Y}$)	η
0.3	76.10%	7.32%
1.2	69.24%	6.85%
0.75	87.21%	1.53%

6.2.3 Retargeting

One of the advantages of physics-based character animation is its ability to synthesize behaviors for novel situations that are not present in the original data. To evaluate our framework's robustness to these discrepancies, we directly copy the local joint rotations from the normal hand (with the finger size factor of 1) to the special-topologies hand model ("long fingers" hand with the size factor of 1.3, "short fingers" hand with the size factor of 0.7, as well as "irregular fingers" hand with the size factor of 0.7 and 1.3) without any further modification.

The grasps generated by the three virtual hand models are shown in Fig.13. Despite the starkly different hand morphologies, our policy is also able to perform well with slight centroid deviation.

(a) "Long fingers" hand

(b) "Short fingers" hand

(c) "Irregular fingers" hand

Fig. 13 Grasps with different hand topologies.

7 Conclusion

We present a novel video-based adaptive grasp planning framework in this paper. Our framework firstly gets reference motion using monocular hand pose estimation, and then obtains the adaptive ability from deep reinforcement learning controller. Our framework can generate natural and stable grasps to different primitive objects. Moreover, users can conveniently customize generic grasps, which can be employed in action demonstrations of virtual human or other interactive scenarios.

Although the experiments illustrate the flexibility and stability of our approach, there are still limitations to be addressed in our future work.

(1) Extend the adaptive capabilities of motion fragments to the entire interactive process. In this work, we clip the grasping sequences, and we can only perform adaptive grasping generalization on the given action primitives. Future work can consider recording hand motion sequences during the whole demonstration process, further exploiting the advantages of inverse reinforcement learning to perform complex interactions.

(2) Apply high-level semantic understanding on the target objects. Since our approach merely considers low-level geometric features, object features such as holes in a bowling ball that are intentionally designed for grasping can't be captured. The combining of the high-dimensional semantic information with our controller will be an interesting direction, so the automatic identification and grasps can be expected.

Acknowledgments

This work was supported by the National Key R&D Program of China (grant number: 2018YFB1004904), the Fundamental Research Funds for the Central Universities under Grant 30918012203.

Declarations

Conflict of interest
The authors declare that they have no conflict of interest.

Data availability
Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Code availability
The code is available at https://zhiyongsu.github.io.

References

1. Antotsiou D, Garcia C, Tae K (2018) Task-oriented hand motion retargeting for dexterous manipulation imitation. In: Computer Vision - ECCV 2018 Workshops, pp 287–301
2. Aravind R, Vikash K, Abhishek G (2018) Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In: Science and Systems. Pittsburgh, pp 1–9
3. Brahmbhatt S, Handa A, Hays J (2019) Contactgrasp: functional multi-finger grasp synthesis from contact. In: IEEE International Conference on Intelligent Robots and Systems, pp 2386–2393
4. Buckingham G (2021) Hand tracking for immersive virtual reality: opportunities and challenges. `https://arxiv.org/abs/1809.02627`
5. Cao Z, Sheikh Y, Simon T (2019) Pose: realtime multi-person 2d pose estimation using part affinity field. IEEE Transactions on Pattern Analysis And Machine Intelligence 35:1–14
6. Chen Y, Tu Z, Ge L (2019) So-handnet: self-organizing network for 3d hand pose estimation with semi-supervised learning. In: The IEEE/CVF International Conference on Computer Vision, pp 6960–6969
7. Ciocarlie M, Goldfeder C, Allen K (2007) Dimensionality reduction for hand-independent dexterous robotic grasping. In: The Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3270–3275
8. Documentation U (2021) Rigidbody. `https://docs.unity3d.com/Manual/class-Rigidbody.html`
9. Ferrari C, Canny J (1992) Planning optimal grasps. In: the IEEE International Conference on Robotics and Automation.Los Alamitos, pp 2290–2295
10. Franziska M, Florian B, Oleksandr S (2018) Generated hands for real-time 3d hand tracking from monocular rgb. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 49–59
11. Hao T, Changbo W, Manocha D (2019) Realtime hand-object interaction using learned grasp space for virtual environments. IEEE Transactions on Visualization and Computer Graphics 25:2623–2635
12. He W, Pirk S, Yumer E (2019) Learning a generative model for multi-step humanobject interactions from videos. In: Computer Graphics Forum, pp 367–378
13. Ji K, Nguyen T, TaeK (2009) 3-d hand motion tracking and gesture recognition using a data glove. In: 2009 IEEE International Symposium on Industrial Electronics, pp 1013–1018
14. Joao P, Thiago M, Thiago L (2020) Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. Computers & Graphics 94:11–21
15. Julia S, Christian E, Simon O (2018) Synergy-based, data-driven generation of object-specific grasps for anthropomorphic hands. In: IEEE-RAS 18th International Conference on Humanoid Robots, pp 327–333
16. Juliani A, Berges V, Vckay E (2019) Unity: a general platform for intelligent agents. `https://arxiv.org/abs/1809.02627`
17. Kanazawa A, Black M, Jacobs D (2018) End-to-end recovery of human shape and pose. In: CVF conference on computer vision and pattern recognition, pp 7122–7131
18. Kopicki M, Adjigble M, Stolkin R (2016) One shot learning and generation of dexterous grasps for novel objects. International Journal of Robotics Research 35:959–976
19. Libin L, Jessica H (2017) Learning to schedule control fragments for physics-based characters using deep q-learning. ACM Transactions on Graphics 37:1–14
20. Libin L, Jessica H (2018) Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. ACM Transactions on Graphics 37:15–21

21. Miller A, Allen P (2004) Graspit! a versatile simulator for robotic grasping. IEEE Robotics & Automation Magazine 11:110–122
22. Min L, Zherong P, Kai X (2019) Generating grasp poses for a high-dof gripper using neural networks. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1518–1525
23. Naijun L, Tao L, Yinghao C (2019) A review of robot manipula- tion skills learning methods. Acta Automatica Sinica 45:458–470
24. Peng s, Zhongqi F, Ligang L (2018) Grasp planning via hand-object geometric fitting. The Visual Computer 34:257–270
25. Roccetti M, Marfia G, Zanichelli M (2010) The art and craft of making the tortellino: Playing with a digital gesture recognizer for preparing pasta culinary recipes. Computers in Entertainment 8:1–20
26. Soshi S, Vladislav G, Weipeng X, Christian T (2020) Physcap: Physically plausible monocular 3d motion capture in real time. ACM Transactions on Graphics 39:1–16
27. Starke J, Eichmann C, Ottenhaus S (2018) Synergy-based, data-driven generation of object- specific grasps for anthropomorphic hands. In: IEEE-RAS International Conference on Humanoid Robots, pp 327–333
28. Weichang C (2014) Comparative analysis and biomechanical analysis of human motion based on kinect. Master's thesis, Tianjin University
29. Xiaoyuan W, Hao T, Changbo W (2020) Research on natural grasp generation of the virtual hand. Journal of Computer-Aided Design & Computer Graphics 32
30. Xiong Z, Qiang L, Hong M (2019) End-to-end hand mesh recovery from a monocular rgb image. In: IEEE International Conference on Computer Vision, pp 2354–2364
31. Xuebin P, Kanazawa A, Malik J (2018) Sfv:reinforcement learning of physical skills from videos. ACM Transactions on Graphics 37:178–192
32. Xuebin P, Sergey L, De V (2018) Deepmimic: example-guided deep reinforcement learning of physics-based character skills. ACM Transactions on Graphics 37:1–14
33. Yili F, Cheng L (2009) Hand modeling and motion controlling based on lay figure in virtual assembly. Computer Integrated Manufacturing Systems 15:681–684
34. Yu R, Park H, lee J (2019) Figure skating simulation from video[j]. computer graphics forum. ACM Transactions on Graphics 38:225–234
35. Yuxiao Z, Marc H, Weipeng X (2020) Monocular real-time hand shape and motion capture using multi-modal data. In: Computer Vision and Pattern Recognition, pp 5346–5355
36. Zicong L, Fanzhong Z, Zihui W (2020) Training a virtual tabletennis player based on reinforcement learning. Journal of Computer-Aided Design & Computer Graphics 32:997–1008