DexFuncGrasp: A Robotic Dexterous Functional Grasp Dataset Constructed from a Cost-Effective Real-Simulation Annotation System

Jinglue Hang, Xiangbo Lin*, Tianqiang Zhu, Xuanheng Li, Rina Wu, Xiaohong Ma, Yi Sun

School of Information and Communication Engineering, Dalian University of Technology, China
{dllgdxhjl0907, zhutq, hswrn}@mail.dlut.edu.cn, {linxbo, xhli, maxh, lslwf}@dlut.edu.cn

Abstract

Robot grasp dataset is the basis of designing the robot's grasp generation model. Compared with the building grasp dataset for Low-DOF grippers, it is harder for High-DOF dexterous robot hand. Most current datasets meet the needs of generating stable grasps, but they are not suitable for dexterous hands to complete human-like functional grasp, such as grasp the handle of a cup or pressing the button of a flashlight, so as to enable robots to complete subsequent functional manipulation action autonomously, and there is no dataset with functional grasp pose annotations at present. This paper develops a unique Cost-Effective Real-Simulation Annotation System by leveraging natural hand's actions. The system is able to capture a functional grasp of a dexterous hand in a simulated environment assisted by human demonstration in real world. By using this system, dexterous grasp data can be collected efficiently as well as cost-effective. Finally, we construct the first dexterous functional grasp dataset with rich pose annotations. A Functional Grasp Synthesis Model is also provided to validate the effectiveness of the proposed system and dataset. Our project page is: https://hjllll.github.io/DFG/.

Introduction

Robotic grasp has the potential to assist humans in performing various operations. Vision-based works of parallel-jaw grippers have already gained wide applications in recent years, however, achieving human-like, task-oriented grasps with high-DoF dexterous robot hand is highly underexplored. These grasps go beyond simple pick-and-place tasks and require not only stable grasp but also making preparation for performing post-grasp manipulation to complete a specific task, such as putting the index finger on the handle to use the spray bottle. For clarity, we refer to this type of grasp as functional grasp, which is the first step towards robot functional manipulation. As shown in Fig.1, functional grasps (top) and stable grasps (bottom) on the same objects are presented and compared.

Currently, one of the challenges in generating dexterous robot functional grasp using deep learning model is the lack of relevant datasets, because annotating dexterous hands with high-DoF is exceptionally difficult. Prior datasets (Liu et al. 2019b; Wei et al. 2022; Wang et al. 2023) obtain stable grasp annotations using the sampling-based method or synthesis method based on differentiable force closure criteria. However, these grasps lack anthropomorphic qualities and do not satisfy human-like functional grasp requirements. (Zhu et al. 2021) firstly provide a dataset related to functional grasp of dexterous hand. They annotate the relationship between the hand and the object with touch codes on the object point clouds, however, the entire process requires manual segmentation of point clouds and then labels each part manually. Conversely, we develop an annotation-efficient method that can collect large amounts of functional grasp poses directly.

For the high-DoF dexterous robot hand dataset annotation system, it is supposed to be as efficient as possible. Previous works commonly use wearable gloves (Liu et al. 2017, 2019a) or haptic sensors (Elsner et al. 2022; Kumar and Todorov 2015) to collect robot actions from human demonstrations, however, the high costs limit their wide

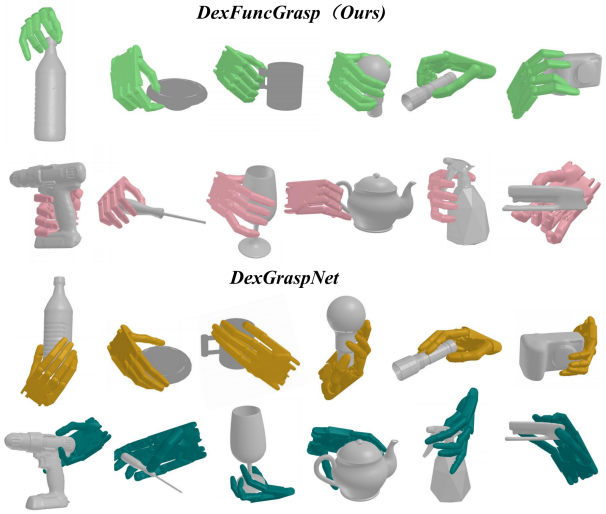

Figure 1: Visualization of our DFG dataset(Dexterous Functional Grasp) and comparison with DexGraspNet(stable grasp).

*Corresponding author. This work was supported by the National Natural Science Fundation of China(61873046, U1708263).

application. (Handa et al. 2020; Arunachalam et al. 2023) use real robots to collect data which cannot satisfy the cost-effective requirement either. Recent developed vision-based teleoperation works (Li et al. 2019; Qin, Su, and Wang 2022) aim to collect robot manipulation trajectories from human demonstrations without providing grasp pose annotations. Motivated by these works, we propose a Real-Simulation Annotation System for functional grasp dataset generation. The Annotation System consists of a simple and cost-effective Real-Simulation scheme. It directly annotates dexterous robot hand grasp poses, contact regions in object, and keeps the dexterous grasp pose as similar as human hand pose. In real world, the human hand poses are estimated from the image sequences and mapped to the simulated dexterous hand, ensuring a consistent action between them. In simulation stage, the dexterous hand is manipulated by a human hand to grasp the function regions of the object in the simulation environment, to collect the grasp poses of dexterous hand as well as the interaction information with object. The proposed Annotation System can obtain plentiful diverse human-like poses in a cost-effective way, including not only stable grasps but also functional grasps making the annotations of dexterous grasp poses on objects an easy task.

By utilizing our Annotation System, we generate the functional grasp dataset DexFuncGrasp (DFG) for dexterous robot hands, containing 559 objects from 12 commonly used categories in daily life and covering over 14k functional grasp pose annotations. In the established dataset, the annotations of each object include the multiple grasp pose annotations, more comprehensive than(Zhu et al. 2021), which only contain contact regions. In addition, we provide an end-to-end Functional Grasp Synthesis Model, simulation and real-world experiments are conducted to demonstrate the effectiveness of our dataset.

Our contributions can be summarized as follows:

- A Real-Simulation Annotation System is proposed for annotating human-like dexterous robot hand grasps. Its data annotation process is cost-effective and convenient assisted by mapping human hand's action to dexterous hand in real time.
- A synthetic dexterous functional grasp dataset (DFG) is established, where each object has multiple grasp pose annotations, contact regions and category label. The rich annotations provide the flexibility of studying diverse functional grasp generation algorithms.
- An end-to-end Functional Grasp Synthesis Model is proposed to evaluate the effectiveness of the dataset. The results of simulation and real experiments prove that the DFG dataset can be used to train a good functional grasp generation model.

Related Work

Dexterous Robot Grasp Datasets

Most existing datasets for dexterous robot hands (Liu et al. 2019b; Goldfeder et al. 2009) adopt the commonly used direct sampling scheme, where a point on the object surface is first sampled, followed by approaching and firmly grasping using the GraspIt!(Miller and Allen 2000) planner. (Wei et al. 2022) improve the grasp diversity by initializing 5 common grasp types, and (Liu et al. 2021) employ the force closure measure as the optimization objective. However, the dexterity of multi-finger hands is not completely covered, while (Wang et al. 2023) fix this problem by sampling better initial grasp and then generating larger datasets. Despite these achievements, the methods and datasets mentioned above are limited in their ability to collect precise or specific grasps such as functional grasps. (Zhu et al. 2021) propose a functional grasp dataset for dexterous multi-fingered robot hand. They use manually annotated "touch code" to represent the contact relationship between hands and objects. However, the approach suffers from the labor-intensive annotation process. The size of their dataset is also limited and lack of ground truth grasp poses.

To address these limitations, we propose a Real-Simulation Annotation System using only consumer cameras and collecting the information in simulation, which has the ability to collect the human-like specific dexterous grasps in a convenient way. To this end, we generate DFG dataset, including functional grasp pose annotations, contact regions and category label. Compared to other datasets shown in Tab.1, our dataset both contains grasp pose annotations and is functional.

Functional Grasp Generation

Compared to generating dexterous stable grasps(Liu et al. 2019b; Wei et al. 2022), functional grasp generation is more challenging due to the lack of datasets and the complexity of some functional grasps, for example, grasp the bottle cap requires the fingertips precisely in contact with the cap. (Brahmbhatt et al. 2019) firstly predict contact information and then takes it as the condition to filter dexterous functional grasps through optimization method. (Zhu et al. 2021) adopt deep learning network to obtain functional grasp based on touch code. (Zhang et al. 2023) further improve the grasp quality and success rate by proposing the attention mechanism based neural network. Similarly, (Xu et al. 2023) use affordance to guide functional grasps by using semantic information. In this work, we also propose an end-to-end Functional Grasp Synthesis Model, and perform simulated and real grasp experiments using the generated functional grasp poses to verify the usability of DFG dataset.

Method

As shown in Fig.2, the proposed Annotation System consists of two parts: Real World and Simulation. The human hand's grasp actions are captured by two RGB cameras and a depth camera simultaneously in Real World and mapped to simulated dexterous robot hand. The robot hand conducts the same action in Simulation in real-time. Through the collaboration between the human hand and the robot hand by consistent action mapping, the robot hand performs grasping of object model. Once the grasp pose is verified successfully in Isaacgym platform(Makoviychuk et al. 2021), the grasp pose annotations, the contact regions and category label will be recorded to construct the dexterous grasp dataset.

ShadowHand is chosen as the default dexterous robot hand in our Annotation System, because it is the most

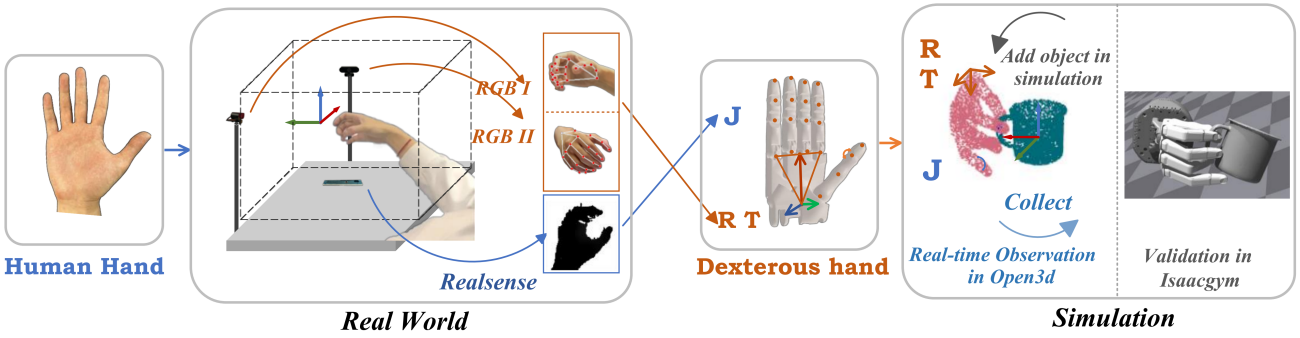

Figure 2: Pipeline of Our Annotation System. The left part sketches the hand pose estimation process in Real World, while the right part sketches the collection process of the dexterous functional grasp pose in Simulation.

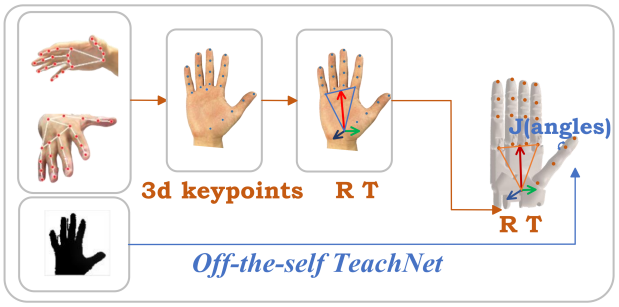

Figure 3: The calculation of dexterous hand parameters from human poses.

dexterous anthropomorphic robot hand with high-DOF currently. Although the ShadowHand is designed to imitate the structure of human hand as much as possible, there are still differences in the size, structure, and kinematics between them. So action retargeting from human hand to dexterous hand should be done for reasonable dexterous grasp. Or else, unexpected finger collision or penetration will occur in performing dexterous grasp. Generally, human hand and dexterous robot hand have different grasp pose representations. 3D coordinates of 21 key points are commonly used for human hand, while the dexterous hand is characterized by its 6D pose R, T and joint angles J. By transforming the representation of human hand to that of the dexterous hand, the action mapping model is established. In the following sections, the scheme details of our Annotation System will be presented.

Hardware and Simulation Setup

Considering the cost-effectiveness, ease of use, real-time performance and technical characteristics comprehensively, three cameras on a single host are used in Real World setup. As shown in Fig.2, one RGB camera is placed in the front of the workspace, while the other RGB camera is placed in the left side, with their optical axes as close to perpendicular as possible. The Realsense SR305 depth camera is placed at the bottom. The workspace is a cube with a side length of 0.5m, where its center is at the intersection of the optical axes of the two RGB cameras. Such configuration can fully cover the movement area of the human hand for a better hand pose observation.

As shown in the rightmost of Fig.2, the Simulation workspace includes two environments. The Open3d(Zhou, Park, and Koltun 2018) is applied for the visualization of the entire dexterous grasp process. The settings of simulation workspace corresponds to the Real World workspace, and the object model is placed at the center of the workspace. The observation perspective is the same in the two workspaces. The robot hand grasp platform IsaacGym is utilized to verify the effectiveness of the grasps. The grasp pose obtained from the Open3d environment is directly given to the dexterous hand in IsaacGym environment. In the simulation environments, the gravity of the object is set according to its category. If the object is stably grasped in mid-air for 10 seconds, the current grasp pose is valid.

Data Acquisition Pipeline

As shown in Fig.2, the data collection using our Annotation System contains two stages, which are estimating the human hand pose in Real World and rebuilding the dexterous robot grasp in Simulation.

First Stage The RGB images captured by two RGB cameras are used to estimate the 3D human hand pose with mediapipe(Lugaresi et al. 2019), which is a light 2d key point detection neural network. To calculate the key point's 3D position from 2D coordinates, the camera's intrinsic parameters (K) and extrinsic parameters ($RT_{c_1 \to c_2}$) are obtained first. The extrinsic parameters is shown in (1), where c1 and c2 represent camera1 and camera2, R and T mean rotation and translation:

$$RT_{c_1 \to c_2} = \begin{bmatrix} R_{c_1 \to c_2} & T_{c_1 \to c_2} \\ 0 & 1 \end{bmatrix} \quad (1)$$

The origin of the world coordinate frame (w) is set at the center of Real World workspace, where the axis is given as shown in Fig.2. The orientations of x-axis (red) and y-axis (green) point to the camera, respectively, while the orientation of z-axis (blue) is upward. Specifically, the x-axis is set

to be parallel to the optical axis of c1, so the transformation matrix from w to c1 is shown in (2):

$$RT_{w\to c_1} = \begin{bmatrix} R_{w\to c_1} & T_{w\to c_1} \\ 0 & 1 \end{bmatrix} \tag{2}$$

As shown in (3), when the key point's 2D coordinates (u_1, v_1),(u_2, v_2) are detected using images from c1 and c2, its 3D coordinate α_{3d} in the coordinate frame of camera1 will be calculated using direct linear transform (DLT) through singular value decomposition (SVD).

$$\begin{bmatrix} v_1 p_{13} - p_{12} \\ p_{11} - u_1 p_{13} \\ v_2 p_{23} - p_{22} \\ p_{21} - u_2 p_{23} \end{bmatrix} \alpha_{3d} = 0 \tag{3}$$

Here P is the projection matrix of each camera calculated from (4), where K is the camera intrinsic.

$$P_1 = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \end{bmatrix} = K_1 \begin{bmatrix} R_{c_1} & T_{c_1} \\ 0 & 1 \end{bmatrix}, P_2 = \begin{bmatrix} p_{21} \\ p_{22} \\ p_{23} \end{bmatrix} = K_2 \begin{bmatrix} R_{c_2} & T_{c_2} \\ 0 & 1 \end{bmatrix} \tag{4}$$

Finally, the key point's 3D coordinates α_{3dw} in the world coordinate frame is obtained using (5).

$$\begin{bmatrix} \alpha_{3dw} \\ 1 \end{bmatrix} = RT^{-1}_{w\to c_1} \begin{bmatrix} \alpha_{3d} \\ 1 \end{bmatrix} \tag{5}$$

To control the dexterous robot hand to perform the same action as human hand, the inferred rotation and translation $[R_{hand}, T_{hand}]$ should be transferred from human hand to robot hand. $[R_{hand}, T_{hand}]$ describes the global movement of the hand, and are obtained from the human hand key point's 3D coordinates α_{3dw} in the world coordinate frame. As shown in Fig.3, the MCP joint of the index finger, the little finger and the wrist joint are selected to form a plane approximately. The origin is set at the wrist joint with the x-axis (red) perpendicular to the line connecting the two MCP joints, y-axis (green) in the plane, pointing towards the thumb, thus the two vectors of our coordinate $(\vec{X}, \vec{Y})$ is first obtained. The z-axis $(\vec{Z})$ (blue) is determined using equation (6), note that they are unit vectors.

$$\vec{Z} = \vec{X} \times \vec{Y} \tag{6}$$

Finally, T_{hand} is set using the positions of the wrist joint, $[R_{hand}, T_{hand}]$ is set as follows:

$$R_{hand} = \begin{bmatrix} \vec{X} \\ \vec{Y} \\ \vec{Z} \end{bmatrix}, T_{hand} = \begin{bmatrix} x_{wrist} \\ y_{wrist} \\ z_{wrist} \end{bmatrix} \tag{7}$$

In order to guarantee clear observations, the human hand moves around the origin of the given world coordinate frame. In Simulation, the object center is set as the origin of the world coordinate frame, and the setting of coordinate axes is the same as that in Real World. Since the object is fixed in Simulation workspace, this setting ensures that the dexterous hand moves around the object, keeping consistency with human hand's movement. Through such settings, the obtained $[R_{hand}, T_{hand}]$ from human hand can serve as the global pose of the dexterous hand directly.

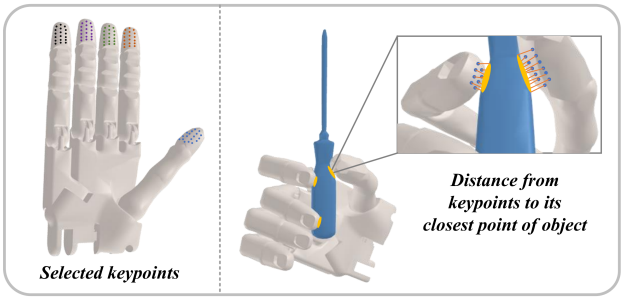

Figure 4: Selected keypoints and the distance calculation.

Second stage Since there is geometric difference between human hand and dexterous robot hand, the human hand pose should be mapped to the robot hand in a proper way. Direct joint angle mapping is a commonly used method, however, it may lead to collision between robot fingers. To overcome this drawback, TeachNet(Li et al. 2019) model is adopted in the proposed Annotation System. It is an off-the-shelf method that can obtain the joint angles J of the ShadowHand from depth images of human hand directly and does not suffer from their domain gap.

After completing the action mapping, the grasp pose [R, T, J] of the dexterous hand will be transferred to the IsaacGym verification platform. The details will be described in the following section.

Verification and Region Annotation

Grasp verification in IsaacGym(Makoviychuk et al. 2021) platform is frequently used in many synthetic datasets, such as (Wang et al. 2023; Xu et al. 2023; Zhang et al. 2023). We place objects in mid-air and reset the collected grasps in the simulator. To simulate realistic conditions, gravity is incorporated into the system. If the object does not rotate more than 20° and is stably grasped by the dexterous hand for 10s, it is considered a successful grasp. Each successful grasp is recorded as an effective data, consisting of grasp pose annotations, the contact regions and category label.

The contact regions provide useful clues for studying grasp pose generation model, grasp pose transfer method, and data augmentation. As shown in Fig.4, since the fingertips play important role in functional grasp, the points of fingertips are used to calculate the contact regions. If the distance between a point of the object surface and any point of a fingertip is less than the threshold (1cm), it is determined to be the contact point.

DexFuncGrasp Dataset

A dataset with diverse annotations of dexterous functional grasp is obtained, which covers 559 instances across 12 categories and more than 14k grasp poses. It is named DexFuncGrasp (DFG) dataset and its production process will be described in this section.

Dataset	Hand	Obj	Method	Grasps	Functional	Pose
DDGdata(Liu et al. 2019b)	ShadowHand	565	Sampling	6.9k	✗	✓
DVGG(Wei et al. 2022)	HIT-DLR II Hand	300	Sampling	1.5M	✗	✓
DexGraspNet(Wang et al. 2023)	ShadowHand	5355	Optimization	1.32M	✗	✓
Toward human-like grasp(Zhu et al. 2021)	ShadowHand	129	Human Annotation	-	✓	✗
DexFuncGrasp(Ours)	ShadowHand	559	Real-Simulation	14k	✓	✓

Table 1: Comparison of our dataset with the publicly available dexterous robot hand datasets.

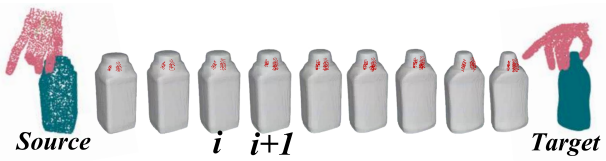

Figure 5: Grasp transfer from one instance to another instance of the same category.

Object Preparation

To reduce the time required for dexterous hand pose annotation, only two instances are selected in each category to be annotated using our Annotation System. Other data in DFG dataset are generated using annotation transferring method (as shown in Fig.5), while the contact regions are transferred through two instances by utilizing Tink method(Yang et al. 2022) and then grasp configuration is optimized.

Functional Grasp Annotation and Dataset Extension

10 subjects participate the functional grasp data collection. Before controlling the simulated dexterous robot hand, they are shown the functional parts of each object, instructed to use the Annotation System. The object is fixed in the simulation environment, and the subjects manipulate the simulated dexterous hand with bare-handed movements in Real World workspace. Each instance requires more than 25 successful grasps. The contact state between the robot hand and the object is observed during the grasp procedure from three different perspectives. Once there are adequate contacts between fingers and object functional regions, the collected grasp is synchronized with IsaacGym for verification. If successful, the annotated grasp poses, the calculated contact regions and the category label are added to DFG dataset.

To augment the dataset, the obtained annotations of the selected two instances are transferred to other instances in the same category. Tink(Yang et al. 2022) is used as the annotation transfer method with the differentiable kinematics model as the dexterous hand model. Let the annotated instance be the 'source', while other instances in the same category be the 'target', 10 intermediate instances are acquired through interpolation between the source instance and the target instance. Since(Park et al. 2019) use an implicit variable code to represent the *sdf* value in object surface reconstruction, where *sdf*(signed distance field) means the distance from the point to the surface of the object, the gradual code interpolation is adopted to reconstruct those intermediate instances, as shown in Fig.5. Then the contact labels are transferred between two adjacent instances from 'source' to 'target', where the corresponding vertices are determined using the iterative closest point (ICP) algorithm. According to the transferred contact regions in the target instance, the [R, T, J] for the 'source' instance is firstly taken as the initial pose for the 'target' instance, then Adam optimizer is used to refine the grasp pose under the guidance of contacting the specified regions with the selected key points shown in Fig.4 of the dexterous robot hand.

The effectiveness of transferred grasps is also verified in IsaacGym platform, only successful grasps can be added into dataset. Through this data augmentation process, the number of data in DFG dataset expands from less than 2000 to over 14000.

Model and Experiments

To verify that the proposed Annotation System is reasonable and the established dataset DFG can be used to develop grasp generation model, a Functional Grasp Synthesis Model is firstly proposed. Then we conduct experiments for validation, note that the DFG dataset is randomly split into training and test sets in a ratio of 9:1, shown in Tab.2. After getting the optimal model, the generated grasp poses are evaluated in both simulation (IsaacGym) and real-world experiments. The details will be presented in the following sections.

Functional Grasp Synthesis Model

The overall architecture of the proposed dexterous functional grasps generation model is shown in Fig.6, whose three sequentially connected modules are partial point clouds completion module, functional grasp generation module, and grasp pose refinement module.

First, as shown in Fig.6①, VRCENT(Pan et al. 2021) model is adopted to achieve the partial point clouds completion and trained on DFG dataset. Then, as shown in Fig.6②, The designed functional generation module is on the basis of the CVAE(Conditional Variational Autoencoders). The object point clouds are served as conditions, and the encoder learns to map [R,T,J] parameters of the dexterous hand into latent code, then taking the object point clouds as condition, z will be decoded to hand parameters through the decoder. Thus the corresponding functional grasps of each category will be generated. The encoder and decoder are based on pointnet++(Qi et al. 2017) network. Finally, as shown in

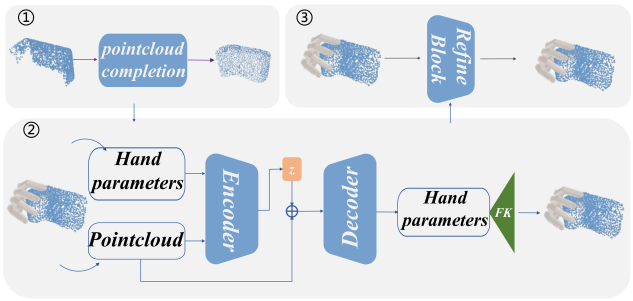

Figure 6: Functional Grasp Synthesis Model which contains point cloud completion module①, a grasp generation module②, and a refinement module③.

Fig.6③, the grasp pose refinement module is embedded at the end of the model, in which the Adam Optimizer is used for further optimizing the grasp pose.

Loss Functions

First, the partial point clouds completion module adopts the chamfer distance loss as the loss function shown in (8), while S_1, S_2 represent two groups of 3D point clouds respectively.

$$L_{cd}(S_1, S_2) = \frac{1}{S_1}\sum_{x\in S_1}\min_{y\in S_2}||x-y||_2^2 + \frac{1}{S_2}\sum_{y\in S_2}\min_{x\in S_1}||y-x||_2^2 \quad (8)$$

Second, to train the functional grasp generation module, the loss function L_{Gen} consists of KL-divergence loss D_{KL} and reconstruction loss (L_v, L_g):

$$L_{Gen} = w_{kl}D_{KL} + w_v L_v + L_g \quad (9)$$

Where D_{KL} is used to force the distribution of the latent code z close to the Gaussian distribution, in which the encoder Q maps the pair of point cloud O and grasp g to latent space z:

$$D_{KL} = -KL(Q(z|O,g), \mathcal{N}(0,\mathbf{I})) \quad (10)$$

Reconstruction loss is used to reconstruct the functional grasp. It consists L_v for optimizing hand mesh vertices and L_g for optimizing the hand parameters r_i, t_i, j_i:

$$L_v = \frac{1}{N}\sum_{i=1}^{N}||\widehat{v_i} - v_i||_1 \quad (11)$$

$$L_g = \frac{1}{N}\sum_{i=1}^{N}(w_r||\widehat{r_i}-r_i||_2^2 + w_t||\widehat{t_i}-t_i||_2^2 + w_j||\widehat{j_i}-j_i||_2^2) \quad (12)$$

where v_i is the sampled vertices on the dexterous hand and N is the number of vertices, w_{kl}, w_v, w_r, w_t and w_j are the trade-off parameters to balance each loss.

Finally, in the refinement module, Adam optimizer is adopted for finetuning the [R, T, J]. As shown in (13), L_{close} and L_{away} are used to force sufficient contact and penalize hand-object penetration. λ_1 and λ_2 are the trade-off parameters to balance each loss.

$$L_{refine} = \lambda_1 L_{away} + \lambda_2 L_{close} \quad (13)$$

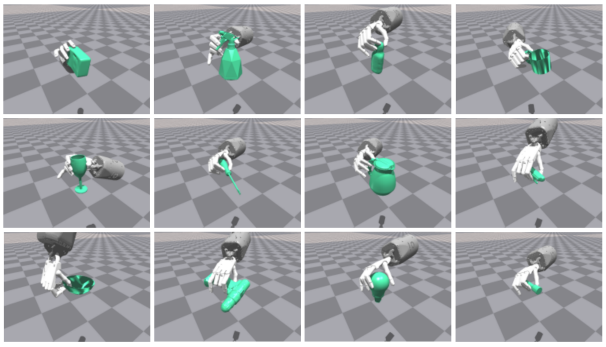

Figure 7: Simulation results in Isaacgym of each category.

Category	SR Ours (FuncGrasp)	CR Ours	Train / Test
Bowl	**69.48%** (-)	85.00%	43 / 5
Lightbulb	**74.23%** (18.75%)	82.29%	44 / 7
Bottle	**68.62%** (0.0%)	71.86%	51 / 8
Flashlight	**91.03%** (31.25%)	90.63%	44 / 6
Screwdriver	**75.60%** (50.0%)	94.27%	32 / 7
Spraybottle	**58.07%** (-)	59.38%	20 / 3
Stapler	**85.77%** (100.0%)	53.91%	28 / 6
Wineglass	**55.70%** (-)	34.38%	35 / 5
Mug	**54.62%** (-)	95.98%	57 / 7
Drill	**55.00%** (-)	61.91%	10 / 2
Camera	**63.33%** (100.0%)	37.50%	87 / 7
Teapot	**57.14%** (-)	98.43%	41 / 7
Total	**67.38%** (50.0%)	**72.13%**	**492 / 67**

Table 2: Comparison of results on test set in simulation experiments. The last column presents the number of objects used for training and test.

$$L_{close} = \sum_{i=1}^{5}\alpha_i\sum_{j=1}^{N}\min(||h_{close}^{ij} - o_{ij}||_2^2) \quad (14)$$

$$L_{away} = \sum_{k=1}^{M}\max(\log\frac{\gamma_k}{||h_{away}^{k} - o_k||_2^2 + \varepsilon}, 0) \quad (15)$$

In (14), h_{close}^{ij} is the selected points in Fig.4 of each finger and o_{ij} is their corresponding closest points on the object, N is the number of sampled points on each fingertip, 5 is the total finger number and α_i is the weight of each finger. In (15), h_{away}^{k} is the unselected points of each finger and o_k is their corresponding closest points on the object. M is the total number of the unselected points, ε is a parameter that prevents the denominator from being zero and γ_k is the distance threshold(1cm) for repulsion.

We set w_{kl}, w_v, w_r, w_t, w_j, λ_1 and λ_2 as 0.1, 1.0, 1.5, 100, 10, 1.0 and 1.0 respectively in the experiments. All experiments are carried out on a desktop with 1 Intel Core™ i7-8700 CPU and 1 NVIDIA RTX 2080Ti GPU with 11GB memory. Note that the modules are trained separately and integrated into the whole Functional Grasp Synthesis Model in the inference stage.

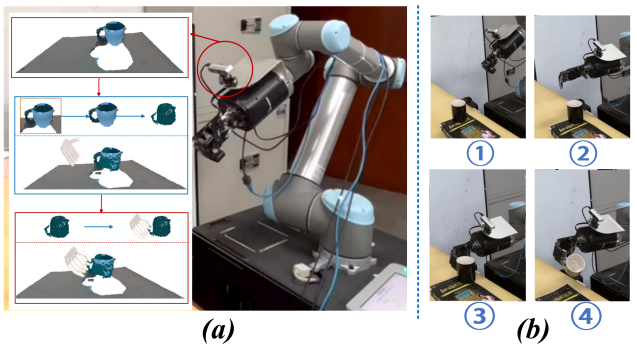

Figure 8: Illustration of real experiment. (a) experimental settings (b) grasp process.

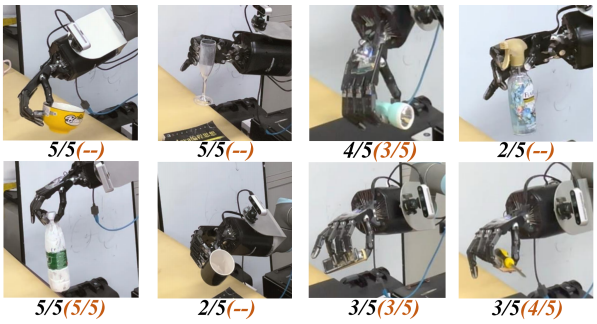

Figure 9: Comparison of success rate between our method and FuncGrasp (in bracket) in real experiments, where the total grasp number of each instance is 5.

Experiments and Results

To evaluate the effectiveness of the DFG dataset and the quality of the generated functional grasp pose from the proposed grasp generation model, the grasp experiments are conducted both in simulation environment and real-world. As there still lacks of approved quantitative evaluation criteria about functional grasp quality, two metrics SR(success rate) and CR (coverage rate) are used. The success rate measures the stability, which calculate the proportion of successful grasps in repeated experiments. We calculate the coverage rate between predicted grasps G_{pre} and ground truth grasp G_{gt} in order to measure the functional grasp quality, once the L_2 distance between them is less than the threshold (t), it is considered as 'coverage', where M is the total number of functional grasps on one object.

$$CR = \frac{1}{M}\sum_{i=1}^{M}\delta(i), \delta(i) = \begin{cases} 1, & ||G^i_{pre} - G^i_{gt}||_2 < t \\ 0, & else \end{cases} \tag{16}$$

Fig.7 shows the qualitative results in simulation. As indicated by the SR and CR values in Tab.2, although the proposed Functional Grasp Synthesis Model is simple, the objects of different categories can be grasped in human-like functional grasp poses, even if the objects in the test set are unseen. The highest success rate can exceed 90%, or 98% on coverage rate. Mug and teapot are two categories that are difficult to be grasped using the generated functional grasp pose, because of the thin handle and the narrow space between the handle and the body. The object is often knocked down when inserting fingers into the handle. So although the generated grasp pose satisfy the functionality well with CR values of 95.98% and 98.43%, their SR values drop down significantly. Considering all categories, the simulation results achieve an average 67.38% Success Rate and 72.13% Coverage Rate, showing that our method is effective in generating functional grasp pose using our DFG dataset.

We also conduct experiments in the real world on real objects with a UR10 arm and a ShadowHand. The process is shown in Fig.8(a), firstly a calibrated Intel Realsense camera is used to obtain RGB and Depth images, then the partial point cloud of the object is obtained through embedded instance segmentation. After that, the object point cloud is taken as input of the trained Functional Grasp Synthesis Model (training by our DFG dataset) to generate the grasp pose. Finally, as shown in Fig.8(b), Moveit! is employed to obtain the motion trajectory for the UR10 arm to complete the final grasp. If the robot lifts up the objects for 10cm and holds it for 10s, it is considered a successful grasp. Fig.9 shows some typical grasp results of one unseen instances from 8 categories. Most categories exhibit high success rates in grasping, while the success rate of grasping Mug is low, similar to that of simulation experiment. The results demonstrate that our dataset and the point-cloud-based Functional Grasp Synthesis Model can effectively generate functional grasps in the real world.

Comparison In order to better show the uniqueness of our work, we first compare it with the DexGraspNet(Wang et al. 2023), a fast grasp generation method, the qualitative comparison results are shown in Fig.1, which proves that our Annotation System obtains reasonable functional grasps, while their method only focuses on stable grasps. Then we compare our method with functional grasp synthesis method FuncGrasp(Zhang et al. 2023), Tab.2 and Fig.9 show the results in simulation and real experiments, respectively. The SR values or the success number of FuncGrasp are taken from their paper. The grasp poses generated by our method achieve SR values of over 50% on all categories, while theirs vary from 0 to 100%, greatly influenced by the object category. It indicates that our dataset can enable a learning-based functional grasp generation model a better performance, even if the model is simpler.

Conclusion and Limitation

In this work, we present a cost-effective and efficient method for collecting high-DoF human-like functional grasp for robot hand. Then a synthetic dexterous functional grasps dataset (DFG) is established through our Annotation System and dataset extension, which avoids the need of tedious manual annotations. Furthermore, we conducted a Functional Grasp Synthesis Model based on DFG and perform simulation and real-world functional grasp experiments, demonstrating the practical applicability of our dataset and method. However, our experiments reveal challenges in achieving precisely grasping cup handles.

References

Arunachalam, S. P.; Güzey, I.; Chintala, S.; and Pinto, L. 2023. Holo-dex: Teaching dexterity with immersive mixed reality. In *IEEE International Conference on Robotics and Automation (ICRA)*, 5962–5969. IEEE.

Brahmbhatt, S.; Handa, A.; Hays, J.; and Fox, D. 2019. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2386–2393. IEEE.

Elsner, J.; Reinerth, G.; Figueredo, L.; Naceri, A.; Walter, U.; and Haddadin, S. 2022. PARTI-A Haptic Virtual Reality Control Station for Model-Mediated Robotic Applications. *Frontiers in Virtual Reality*, 3: 925794.

Goldfeder, C.; Ciocarlie, M.; Dang, H.; and Allen, P. K. 2009. The Columbia Grasp Database. In *IEEE international conference on robotics and automation(ICRA)*, 1710–1716. IEEE.

Handa, A.; Van Wyk, K.; Yang, W.; Liang, J.; Chao, Y.-W.; Wan, Q.; Birchfield, S.; Ratliff, N.; and Fox, D. 2020. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *IEEE International Conference on Robotics and Automation (ICRA)*, 9164–9170. IEEE.

Kumar, V.; and Todorov, E. 2015. Mujoco haptix: A virtual reality system for hand manipulation. In *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 657–663. IEEE.

Li, S.; Ma, X.; Liang, H.; Görner, M.; Ruppel, P.; Fang, B.; Sun, F.; and Zhang, J. 2019. Vision-based teleoperation of Shadow dexterous hand using end-to-end deep neural network. In *International Conference on Robotics and Automation (ICRA)*, 416–422. IEEE.

Liu, H.; Xie, X.; Millar, M.; Edmonds, M.; Gao, F.; Zhu, Y.; Santos, V. J.; Rothrock, B.; and Zhu, S.-C. 2017. A glove-based system for studying hand-object manipulation via joint pose and force sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6617–6624. IEEE.

Liu, H.; Zhang, Z.; Xie, X.; Zhu, Y.; Liu, Y.; Wang, Y.; and Zhu, S.-C. 2019a. High-fidelity grasping in virtual reality using a glove-based system. In *IEEE International conference on Robotics and Automation (ICRA)*, 5180–5186. IEEE.

Liu, M.; Pan, Z.; Xu, K.; Ganguly, K.; and Manocha, D. 2019b. Generating grasp poses for a high-dof gripper using neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1518–1525. IEEE.

Liu, T.; Liu, Z.; Jiao, Z.; Zhu, Y.; and Zhu, S.-C. 2021. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters(RAL)*, 7(1): 470–477.

Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.-L.; Yong, M. G.; Lee, J.; et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Makoviychuk, V.; Wawrzyniak, L.; Guo, Y.; et al. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. *NeurIPS Datasets and Benchmarks*.

Miller, A. T.; and Allen, P. K. 2000. GraspIt!: A Versatile Simulator for Grasp Analysis. *ASME International Mechanical Engineering Congress and Exposition (IMECE)*, 1251–1258.

Pan, L.; Chen, X.; Cai, Z.; Zhang, J.; Zhao, H.; Yi, S.; and Liu, Z. 2021. Variational relational point completion network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, 8524–8533.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, 165–174.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NIPS)*, 30.

Qin, Y.; Su, H.; and Wang, X. 2022. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 7(4): 10873–10881.

Wang, R.; Zhang, J.; Chen, J.; Xu, Y.; Li, P.; Liu, T.; and Wang, H. 2023. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 11359–11366. IEEE.

Wei, W.; Li, D.; Wang, P.; Li, Y.; Li, W.; Luo, Y.; and Zhong, J. 2022. DVGG: Deep Variational Grasp Generation for Dextrous Manipulation. *IEEE Robotics and Automation Letters(RAL)*, 7(2): 1659–1666.

Xu, Y.; Wan, W.; Zhang, J.; Liu, H.; Shan, Z.; Shen, H.; Wang, R.; Geng, H.; Weng, Y.; Chen, J.; et al. 2023. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 4737–4746.

Yang, L.; Li, K.; Zhan, X.; Wu, F.; Xu, A.; Liu, L.; and Lu, C. 2022. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 20953–20962.

Zhang, Y.; Hang, J.; Zhu, T.; Lin, X.; Wu, R.; Peng, W.; Tian, D.; and Sun, Y. 2023. FunctionalGrasp: Learning Functional Grasp for Robots via Semantic Hand-Object Representation. *IEEE Robotics and Automation Letters(RAL)*.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.

Zhu, T.; Wu, R.; Lin, X.; and Sun, Y. 2021. Toward Human-Like Grasp: Dexterous Grasping via Semantic Representation of Object-Hand. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15741–15751.