Neural Fields for Hand-object Interactions

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computer Science and Engineering** by Research*

by

Chandradeep Pokhariya
2021701040
chandradeep.pokhariya@research.iiit.ac.in

Advisor: Dr. Avinash Sharma
Co-Advisor: Dr. Srinath Sridhar

International Institute of Information Technology Hyderabad
(Deemed to be University)
Hyderabad 500 032, INDIA

June 2024

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Neural Fields for Hand-object Interactions*** by *Chandradeep Pokhariya* has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Dr. Avinash Sharma	Co-Advisor: Dr. Srinath Sridhar

To those who believe in themselves...

Acknowledgements

The work presented in this thesis would not have been possible without the immense support of my advisors, family, peers, and friends.

First and foremost, I would like to express my deepest gratitude to my advisor, *Dr. Avinash Sharma.* Despite my lack of a background in computer science, he graciously accepted me first as a research assistant and later as an M.S. student. His unwavering support and invaluable guidance from day one have significantly shaped my research career and life. His saying, "The first paper shouldn't be accepted on the first attempt," has been particularly impactful, emphasizing the importance of honing all the skills necessary for good research. I also appreciate his sponsorship, allowing me to travel and present my research at conferences even with limited funds.

I also wish to extend my sincere thanks to my co-advisor, *Dr. Srinath Sridhar*, for his invaluable research advice on my projects. His kindness in offering me a research scholar position at Brown University last summer gave me immense exposure to international research. His emphasis on creative and fundamental thinking has been truly inspiring.

I am deeply grateful to my family, without whom none of this would have been possible. Despite coming from a remote and small town, my parents were always committed to providing me with the best education. They faced numerous challenges but never let them affect us, continually motivating me with their resilience and dedication. When people ask how I am disciplined and can work so hard consistently, I always tell them they should have seen my parents. I would also like to thank my elder sister *Garima*, who has been by my side for as long as I can remember. She has always motivated me to aim higher and believed in my ability to achieve great things. Additionally, I am thankful to my brother-in-law *Kamal* for organizing numerous trips and hikes, which have been invaluable for relieving the stress and heat of research.

I would like to thank my best friend *Rajesh (bhand* :P) for always inspiring me and supporting me through my highs and lows. Our philosophical discussions about life, people, the world, and the universe have been incredibly engaging and motivating. His way of life truly inspires me. With *Rohit* and *Dhruv*, our friendship has grown since our school days, and I am thankful to them for all the road trips we take when I am back in my hometown.

My heartfelt appreciation goes to my peers and colleagues at CVIT. I am deeply grateful to *Astitva*, who has been a wonderful friend despite not being from my state (:P). His critical thinking skills and creativity have taught me a great deal. *Shanthika*, the most hardworking girl at CVIT, has inspired me with her bold decisions and ambitions. Our midnight discussions about new problem statements

in geometry processing have been like research snacks for me. My gratitude also goes to *Aakash* and *Dhawal* for our insightful conversations about computer graphics research. I am thankful to *Sagar* sir, with whom I worked on my first project in 3D computer vision, gaining substantial knowledge about research in 3D. *Ishaan* and *Rahul*, whom I consider my younger brothers and among the smartest at CVIT, have always been there for making me aware about new technologies and, of course, Gen Z trends (:P). *Amogh*, for being the 'mom' of CVIT and caring for us all. I also thank to *Ritam*, whom I regard as an elder brother (he is quite old, :P), for his help in understanding complex mathematical and physics topics. I am also grateful to everyone else who made my time at CVIT enjoyable, including *Sarath, Surbhi, Sai, Jeet, Neha, Deepti, Pranav, Aparna, Ayan, BVK*, and *Vishal*.

Lastly, I appreciate the members of the IVL at Brown University, especially *Sudarshan, Rahul, Angela, Zekun, Ashish, Rao, Chaerin, Hongyu* and *Arthur*, for their collaboration and engaging research discussions over delicious lunches at IVL meetings.

Abstract

The hand is the most commonly used body part for interacting with our three-dimensional world. While it may seem ordinary, replicating hand movements with robots or in virtual/augmented reality is highly complex. Research on how hands interact with objects is crucial for advancing robotics, virtual reality, and human-computer interaction. Understanding hand movements and manipulation is critical to creating more intuitive and responsive technologies, which can significantly improve accuracy, efficiency, and scalability in various industries. Despite extensive research, programming robots to mimic human-hand interactions remains a challenging goal.

One of the biggest challenges is collecting accurate 3D data for hand-object grasping. This process is complicated because of the hand's flexibility and how hands and objects occlude in grasping poses. Collecting such data often requires expensive and sophisticated setups. However, recently, neural fields [1] have emerged, which can model 3D scenes using only multi-view images or videos. Neural fields use a continuous neural function to represent 3D scenes without needing 3D ground truth data, relying instead on differentiable rendering and multi-view photometric loss. With growing interest, these methods are becoming faster, more efficient, and better at modeling complex scenes.

This thesis explores how neural fields can address two specific subproblems in hand-object interaction research. The first problem is generating novel grasps, which means predicting the final grasp pose of a hand based on its initial position and the object's shape and location. The challenge is creating a generative model that can predict accurate grasp poses using only multi-view videos without 3D ground truth data. To solve this, we developed RealGrasper, a generative model that learns to predict grasp poses from multi-view data using photometric loss and other regularizations. The second problem is accurately capturing grasp poses and extracting contact points from multi-view videos. Current methods use the MANO model [2], which approximates hand shapes but lacks the details for precise contacts. Additionally, there is no easy way to get ground truth data for evaluating contact quality. To address this, we propose MANUS, a method for markerless grasp capture using articulated 3D Gaussians that reconstructs high-fidelity hand models from multi-view videos. We also created a large dataset, MANUS-Grasps, which includes multi-view videos of three subjects grasping over 30 objects. Furthermore, we developed a new way to capture and evaluate contacts, providing a contact metric for better assessment.

We thoroughly evaluated our methods through detailed experiments, ablations, and comparisons, demonstrating that our approach outperforms existing state-of-the-art methods. We also summarize our

contributions and discuss potential future directions in this field. We believe this thesis will help advance the research community further.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advancements in 3D computer vision and computer graphics are transforming various industries by increasing their accuracy, efficiency, and scalability. These fields are evolving rapidly due to the increasing need to better understand and interact with the three-dimensional world. In robotics, 3D computer vision has been revolutionary, enhancing the intelligence and functionality of robots by enabling them to interact more intuitively with their environments. This has led to significant progress in autonomous robots, automated manufacturing, and human-robot interaction. Furthermore, integrating 3D computer vision with Augmented Reality (AR) and Virtual Reality (VR) is paving the way for immersive experiences. These technologies employ real-time 3D mapping and object recognition to foster richer, more interactive digital experiences. These developments underscore the crucial role of 3D computer vision in understanding our complex world and enhancing our interactions with technology. Ongoing advancements in this field broaden robot's ability to perceive, engage with, and influence our three-dimensional surroundings, opening up exciting new opportunities.

A key factor in these developments is the evolution of 3D representations. Traditional 3D modeling techniques, which relied on structured grids, point clouds, or meshes, have given way to novel approaches such as neural fields, which use continuous functions to represent 3D scenes. These methods have recently gained popularity for various visual computing challenges, like reconstructing 3D

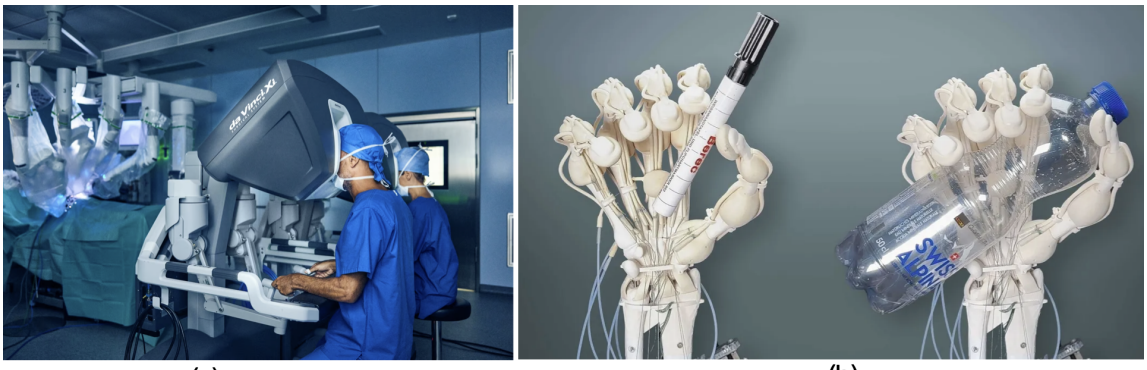

Figure 1.1: Practical use cases of 3D computer vision in (a) robot-assisted surgery (b) robot grasping.

shapes and appearances, generating new viewpoints from existing images, modeling human figures, and enhancing medical imaging. Neural fields are also being applied in fields beyond visual computing, including physics and engineering.

This thesis explores the use of neural fields in 3D computer vision, which addresses the shortcomings of traditional 3D representations and offers capabilities not possible with older methods. Specifically, it focuses on applying neural fields to hand-object interaction problems.

1.1 Motivation and Challenges

Imagine the simple acts of picking up a morning coffee cup, scribbling down a quick note, or tightening a loose faucet with a wrench. These everyday actions don't just show how adept we are at handling different objects; they also present a significant challenge for machines: understanding the nuanced ways humans interact with the physical world.

This challenge isn't just academic; it has practical applications in many areas. For instance, getting this right in robotics is essential for developing advanced technologies that enable robots to carry out complex tasks like assembly, cooking, or even conducting surgeries Figure 1.1. These robots need to do more than grab things—they must handle objects with the precision and care that a human hand can. In augmented reality (AR) and virtual reality (VR), making hand-object interactions more realistic can greatly improve how immersive and interactive these technologies feel. This is especially useful in educational settings, where virtual models can help enhance learning or in professional design, allowing users to work with digital prototypes as if they were real. Moreover, understanding these interactions can lead to better human-computer interfaces, making technology more intuitive and accessible, particularly for people with disabilities. This can range from simple gesture-based controls for everyday tech to more complex interactions in specialized software, improving how users engage with digital systems.

Capturing 3D data, especially during hand-object interactions, presents substantial difficulties that often require expensive equipment and complex setups. The unique challenges include the hand's flexible and non-rigid nature, occlusions caused by the interaction with objects, and the fingers occluding each other. High-end 3D scanners like the Artec Leo Figure 1.2 can generate detailed and textured meshes but are limited to capturing rigid structures. RGBD sensors like Microsoft's Azure Kinect and Intel's RealSense offer a more affordable solution for capturing multi-view 3D hand-object interaction data; however, these sensors face issues like sensor noise, sensitivity to change in light, and precise depth calibration. Their output is typically a sparse and noisy point cloud, and achieving accurate, good-quality geometry is another research challenge. Costly motion capture (MoCap) setups are used to track hand movements in hand-object interactions, which track the hand's movements through markers. While this method is accurate for tracking, it only captures a limited number of points, and fitting a parametric model like MANO [2] to these points to reconstruct the hand's geometry can be imprecise. At the higher end, systems like 3DMD and The Relightables offer robust and high-fidelity 4D data capture, but these are exceptionally expensive and complex, making them hard to mimic with lower-

Figure 1.2: Different capture devices available in the market.

end devices. Additionally, in hand-object interactions, another issue arises: even if accurate 3D data is captured, determining the contact area between the hand and object is challenging because it requires disentanglement between the two. To address this, specialized hardware like heat-sensitive cameras has been employed to capture the contact details in hand-object interactions more precisely; however, these methods also struggle with heat dissipation.

On the representation front, neural fields have shown great potential in encoding scene properties using only multi-view images and videos. The widespread availability of high-quality smartphone cameras simplifies the capture of high-quality 2D images and videos, reducing dependence on error-prone manual 3D annotation methods and democratizing data acquisition. Given these advancements, it's logical to explore the use of neural fields to tackle these existing challenges in the field, leveraging multi-view images and videos of hand-object interactions.

Despite the critical importance, this problem is not trivial at all and comes with multiple challenges which we discuss below,

- **Complexity of Hand Movements**: The human hand is an incredibly intricate system comprising many bones, muscles, tendons, and ligaments. This complexity allows for a vast range of movements and poses, making it challenging to model in 3D accurately Figure 1.4. Capturing the nuances of finger movements, joint rotations, and subtle gestures adds layers of complexity to the modeling process. Additionally, the variability in hand shapes and sizes among individuals further complicates the task.

- **Limited Dataset**: Acquiring precise 3D data on hand-object interactions poses significant challenges due to occlusions caused by hands and fingers. While datasets such as GRAB [3], DexYCB [4] and ARCTIC [5] have contributed significantly to progress in this area, they rely on parametric models like MANO [2], which provide only an approximate depiction of hands lacking intricate details. Consequently, datasets are urgently needed to address these limitations and provide more comprehensive information.

- **Variability in Object Shapes**: Objects exhibit a wide range of shapes, sizes, and textures, further complicating hand-object interaction modeling Figure 1.3. From simple geometric shapes to

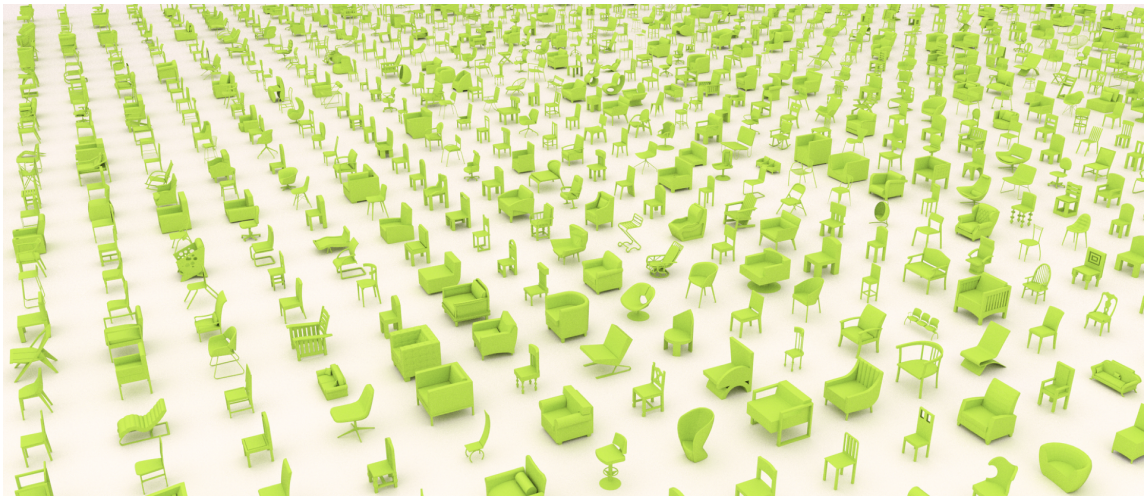

Figure 1.3: Object variability shown in ShapeNet dataset [6]. Image Source.

complex, irregular forms, objects present diverse challenges for interaction strategies. Adapting to this variability requires robust algorithms capable of recognizing and interacting with objects of varying characteristics. Moreover, the material properties of objects, such as hardness, elasticity, and friction, influence how hands interact with them, adding another layer of complexity to the modeling process.

- **Occlusions**: Significant occlusions often occur during hand-object interactions. Fingers or parts of the hand can obscure the object and vice versa, making it difficult to determine the hand pose relative to the object precisely. Vision systems need to be able to handle these occlusions and infer the hidden hand configuration based on the visible portions.

- **Real-time Processing Requirements**: Many applications of hand-object interaction systems, such as robotics and augmented reality, demand real-time processing capabilities. Achieving real-time performance requires algorithms that can efficiently process large volumes of data and make rapid decisions. This necessitates the development of optimized algorithms and computational techniques tailored to the specific requirements of real-time applications. Balancing accuracy with computational efficiency becomes crucial in such scenarios to ensure smooth and responsive interaction experiences.

1.2 Problem Statement

As discussed, given the advantages of the neural field methods, we want to explore them for understanding and modeling of 3D hand-object interactions using real-world multi-view images/videos. We mainly focus on the following two sub-problems in the hand-object interaction domain:

Figure 1.4: Diverse and complexity in hand movements.

Novel Grasp Generation The primary objective here is to synthesize a novel grasp based on the initial pose of a hand and an object, utilizing solely 2D multi-view grasp data. This requires constructing a detailed 3D model of both the hand and the object, followed by developing a grasping framework that accurately predicts a hand pose capable of successfully interacting with the object. The grasping dynamics change significantly with changes in the object's orientation, shape, and size, so the grasping framework must learn these subtle dynamics for accurate grasp generation. This process must be achieved entirely by analyzing multi-view grasping videos, thereby eliminating the need for 3D annotated data.

Markerless Grasp Capture This sub-problem focuses on reconstructing the hand's articulated movements and accurately depicting the sequence of grasps from 2D multi-view videos. The challenge lies in transforming 2D video data into a 3D sequence that accurately represents the physical interactions between the hand and the object, ensuring precise contact modeling throughout the sequence. Furthermore, since there is no ground truth contact data to compare the grasp, there is a strong need in the literature to find methodologies to acquire ground truth contact data and compare with it.

1.3 Research Landscape

1.3.1 Novel Grasp Generation

Novel grasp generation techniques can be categorized into two main groups: physics-based simulations and perception-based methods.

Physics-based simulation methods utilize simulations to model human grasp in synthetic environments due to the challenges of capturing real-world data. One such method is Graspit [7], which employs heuristics and physics principles to determine hand poses based on object interactions. Recent approaches like D-Grasp [8] and Grasp'D [9] enhance dynamic grasp synthesis through reinforcement learning and differentiable simulation, respectively. ManipNet [10] combines marker-based motion capture with machine learning to simulate object manipulations. Despite their advancements, these simulation-based approaches often struggle to bridge the gap between synthetic and real-world data.

Perception-based methods leverage visual data to generate novel grasps, incorporating physical constraints such as proximity, forces, and contacts during optimization or learning processes. Grasping

Field [11] employs the supervised training of a variational auto-encoder, which takes the point cloud of an object and hand pose as input and outputs the grasping pose of the hand. ContactDB [12] uses thermal imaging data to train a model that predicts grasp contact based on the new object shape. Other works, like [13], aim to solve the more ill-posed problem of hand-object reconstruction from RGB videos. While simulation-based grasping techniques cannot be directly extended to real-world data due to the complexity and noise inherent in such data, perception-based methods can work with real-world data but often require extensive 3D annotated data to learn and generate novel grasping dynamics effectively. Approaches like FLEX [14] leverage full-body pose and hand-grasping priors, composing them into 3D geometrical constraints to obtain full-body grasps.

Additionally, all the aforementioned methods rely on parametric hand models like MANO [2], which is an approximation of the actual hand.

1.3.2 Markerless Grasp Capture

Multiple methods attempt to capture hand-object interaction from either single or multi-view sequences. Methods addressing the more ill-posed problem, such as those using a single view, [12], [15] rely heavily on supervised training and require accurate 3D annotated data. The accuracy of these methods does not match those using multi-view camera setups due to fewer visual cues and the occlusion ambiguity that arises during hand-object interaction.

Multiview camera methods still use MANO as the core hand model because it simplifies contact estimation. However, due to the limited expressivity of parametric hand models, the interaction and extracted contacts are often suboptimal. Additionally, the dataset used for training these methods doesn't have enough dense views to avoid occlusion ambiguity. Some methods, like [16], rely on sparse multi-view RGBD data to tackle this problem, although depth information alone does not fully address these challenges.

1.4 Thesis Contributions

- **RealGrasper: Learning Human Hand Grasping from Multi-View Images:** In this work, we introduce a method for modeling human hand grasps using real-world, multi-view image data. We develop neural field representations for both objects and hands, capturing their shapes and appearances. Finally, we devise a generative model named RealGrasper, which generates the final pose of a hand grasp given an object and an initial hand pose.

- **MANUS: Markerless Grasp Capture using Articulated 3D Gaussians:** We present MANUS, a novel method for grasp capture that utilizes an articulated 3D Gaussian representation to model hand shapes with high fidelity. It employs 3D Gaussians to articulate the complexities of hand movements. Leveraging Gaussian primitives that are optimized based on multi-view, pixel-aligned losses, our approach enhances the efficiency and precision of estimating contact points

between the hand and objects. Comparative evaluations against ground truth data demonstrate that our method surpasses conventional template-based methods regarding contact estimation accuracy.

1.5 Thesis Roadmap

This chapter introduced the problem of accurately modeling hand-object interactions and its associated challenges. We also discussed existing state-of-the-art methods and their limitations and briefly mentioned potential solutions to tackle those limitations. In Chapter 2, we provide the necessary background for this thesis and briefly summarize the aspects of various representations. We also discussed the parametric hand model MANO, fitting it to multi-view images, and the inverse kinematics pipeline, which doesn't use the MANO parametric model. In Chapter 3, we explain our novel grasp generation work, Realgrasper. In Chatper-4, we explain grasp capture work MANUS followed by conclusion and future work in late chapters.

Chapter 2

Background

In this chapter, we build a basic background by reviewing useful definitions and terminologies. First, we discuss the neural field representation, especially the Neural Radiance Field and its articulated variants, and then Gaussian Splatting. Then, we also talk about parametric hand representation MANO. We also discuss estimating the hand pose from the multi-view images, including MANO fitting and Inverse Kinematics.

2.1 Neural Fields

2.1.1 NeRF

Neural Radiance Field [17] (NeRF) Figure 2.1 is a technique for reconstructing a 3D scene and generating novel views from a set of 2D images. The method has gained significant traction in computer graphics and computer vision due to its ability to produce high-fidelity images and its applicability across various domains like virtual reality, augmented reality, and movie production.

NeRF represents a scene using a 5D vector-valued function approximated by a Multi-Layer Perceptron (MLP) F_Θ. The input to this network is a 5D vector (x, y, z, θ, ϕ) consisting of 3D spatial coordinates $\mathbf{x} = (x, y, z)$ and a 2D viewing direction vector $\mathbf{d} = (\theta, \phi)$. The MLP maps this input to

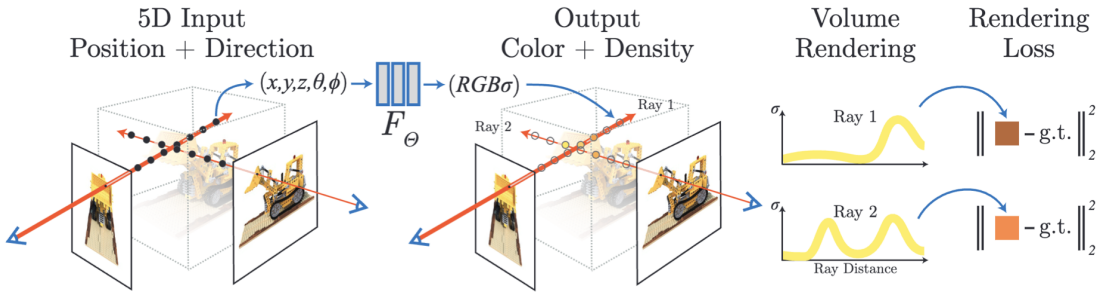

Figure 2.1: Different capture devices available in the market.

output as an RGB color value $(c) = (r, g, b)$ and a volume density σ.

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma) \tag{2.1}$$

During training, rays are cast from the camera center towards each pixel in the image plane. Along each ray, a set of 3D points are sampled by uniform or hierarchical sampling. The 3D coordinates of these sampled points with the θ and ϕ are fed into the MLP. The network predicts the sampled points' volume density σ and color $\mathbf{c}$. The density is a function of only position, while color is a function of both position and viewing direction. Once the color and density of the sampled points are obtained, a volume rendering function is employed to compute the final color for each pixel, which generates the final color of the target pixel by integrating the color and density along the camera ray.

$$C(r) = \sum_{i=1}^{N} T_i \alpha_i c_i \tag{2.2}$$

where,

$$T_i = exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \tag{2.3}$$

Here, T_i is the accumulated opacity, α_i and c_i is the transparency and color of the ith sample.

By minimizing the rendering loss between the predicted images and the ground truth images from different viewpoints, the MLP parameters are optimized.

Furthermore, NeRF incorporates positional encoding to enhance the quality of the rendered images.

Articulated NeRF:

While vanilla NeRF and its variants excel at rendering static scenes, their inability to capture dynamic scenes like articulated objects and characters poses a significant limitation. There have been multiple attempts in the literature to address this challenge. One such approach is Template-Free Animatable Neural Radiance Fields (TAVA) [18].

TAVA represents the shape and appearance of the canonical skeleton. The Linear Blend Skinning (LBS) function is used to get the pose in the world space. Formally, if skinning weights are defined as $\mathbf{w} = (w_1, w_2, ..., w_B, w_{bg}) \in \mathbb{R}^{B+1}$ in the canonical space, and for a given pose $\mathbf{P} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_B\} \in \mathbb{R}^{4\times4}$, forward Linear Blend Skinning (LBS) is used to determine the deformation of a point $\mathbf{x}_c$ in the canonical space to $\mathbf{x}_v$ in the world space. Here, B is the number of bones and, T is per-bone transformation.

$$\mathbf{x}_v = \text{LBS}(\mathbf{w}(\mathbf{x}_c; \Theta_s), \mathbf{P}, \mathbf{x}_c) = \left[\sum_{j=1}^{B} w_j(\mathbf{x}_c; \Theta_s) \cdot \mathbf{T}_j + w_{bg} \cdot \mathbf{I}_d\right] \mathbf{x}_c \tag{2.4}$$

where $\mathbf{I}_d \in \mathbb{R}^{4\times4}$ is the identity matrix. $w_{bg} \cdot \mathbf{I}_d$ term is used to map entire 3D space by assigning the background points and identity transformation. This allows points in the background and empty space to remain unaffected by the skeleton deformation.

To capture the non-linear deformations which are not handled by LBS, an additional term is introduced $\mathcal{F}_\Delta : (\mathbf{x}_c, \mathbf{P}) \rightarrow \Delta_w \in \mathbb{R}^3$ on top of the learned LBS.

The final equation is given by,

$$\mathbf{x}_v = \text{LBS}(\mathbf{w}(\mathbf{x}_c; \Theta_s), \mathbf{P}, \mathbf{x}_c) + \Delta_w(\mathbf{x}_c, \mathbf{P}; \Theta_\Delta). \tag{2.5}$$

Color and density are queried from the canonical space to render this deformed skeleton. This requires finding the corresponding $\mathbf{x}_c$ in the canonical space for each $\mathbf{x}_v$ in the view space. There is no analytical solution to find this correspondence, hence it is posed as root-finding problem and solved using Newton's method.

$$\text{Find } \mathbf{x}_c^*, \text{ such that } f(\mathbf{x}_c^*) = \text{LBS}(\mathbf{w}(\mathbf{x}_c^*; \Theta_s), \mathbf{P}, \mathbf{x}_c^*) + \Delta_w(\mathbf{x}_c^*, \mathbf{P}; \Theta_\Delta) - \mathbf{x}_v = 0 \tag{2.6}$$

$$\mathbf{x}_c^{(k+1)} = \mathbf{x}_c^{(k)} - (J^{(k)})^{-1} f(\mathbf{x}_c^{(k)}), \tag{2.7}$$

where $J^{(k)} \in \mathbb{R}^{3\times3}$ is the Jacobian of $f(\mathbf{x}_c^{(k)})$ at the k-th step.

2.1.2 Gaussian Splatting

In contrast to NeRF's implicit scene representation, Gaussian Splatting [19] employs an explicit approach using 3D Gaussians as building blocks to represent the scene as a point cloud. Each Gaussian is defined by a center point, denoted by μ, representing its mean location, and a covariance matrix, Σ, which captures its shape and orientation. Formally, it is expressed as,

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{2.8}$$

The covariance matrix is further broken down into rotation and scaling matrix as,

$$\Sigma = RSS^T R^T \tag{2.9}$$

Here, each Gaussian primitive has 3D position (μ), opacity, anisotropic covariance matrix (Σ), and spherical harmonics (SH) coefficients.

A tile-based differentiable rasterizer is used to render the Gaussian primitive. The covariance matrix is transformed into camera coordinates using the viewing transformation matrix W and Jacobian matrix J of an affine approximation of the projective transformation.

$$\Sigma' = JW\Sigma W^T J^T \tag{2.10}$$

To determine the final color for a pixel, the contributions of multiple N Gaussians are blended as,

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i) \tag{2.11}$$

Here c_i and α_i represent the density and color of this point computed by a 3D Gaussian G.

2.2 Parametric Hand Model (MANO)

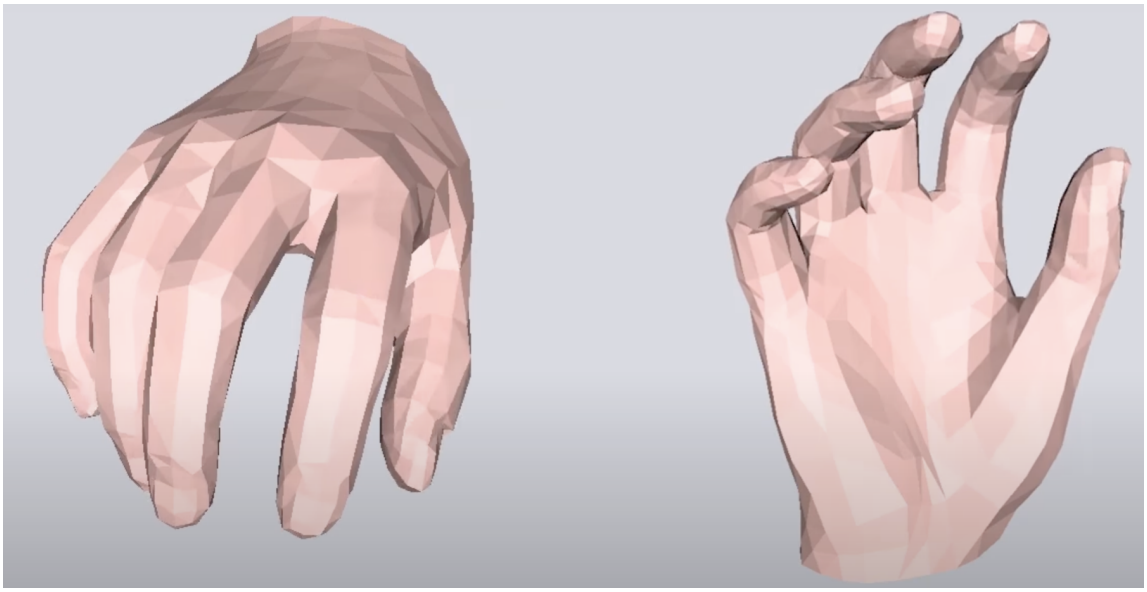

Figure 2.2: Parametric hand model MANO.

MANO [2] is a statistical model Figure 2.2 of the human hand extensively used in computer vision and graphics. It deforms the vertices by blending pose and shape to represent hand shapes accurately and pose. MANO employs a low-dimensional shape space to capture the variability in hand shapes among individuals. Hand pose is represented by a set of joint angles, which control the rotation of the hand's skeletal structure. MANO uses blend shapes to capture the non-rigid deformations of the hand's surface in various poses. It takes pose parameters $\theta \in \mathbb{R}^{48}$ and shape parameters $\beta \in \mathbb{R}^{10}$ as input and outputs a hand mesh with $M \in \mathbb{R}^{V \times 3}$ where V is 778 vertices. Additionally, MANO's joint regressor returns the joints $J \in \mathbb{R}^{K \times 3}$ of the hand for a total of $K = 16$ joints.

$$M(\beta, \theta) = W(T_p(\beta, \theta), J(\beta), \theta, \mathcal{W}) \tag{2.12}$$

$$T_p(\beta, \theta) = \mathbf{T} + B_S(\beta) + B_p(\theta) \tag{2.13}$$

Here, W is a skinning function: Linear Blend Skinning. T_p is posed template, J is joint locations defining a kinetic tree and $\mathcal{W}$ are blend weights.

Specifically, the pose and shape blend shapes are defined as the linear combination of a set of deformations, i.e., vertex offsets.

$$B_p(\theta, \mathcal{P}) = \sum_{n=1}^{9K} (R_n(\theta) - R_n(\theta^*))P_n \tag{2.14}$$

$$B_S(\beta, \mathcal{S}) = \sum_{n=1}^{|\beta|} \beta_n S_n \tag{2.15}$$

Here, $P_n \in \mathcal{P}$ are the pose blend shapes, and K is the number of joints in the hand model.

2.3 Hand Pose Estimation

This section explores the task of 3D hand pose estimation, which involves determining the spatial location and orientation of the hand in the world space. We will discuss two broad categories of methods employed for this purpose.

2.3.1 MANO-based pose estimation

Pose estimation of hands using MANO is an active research area with various problem configurations. These include single and multi-view image hand pose estimation and pose estimation from 3D meshes. Here, we will specifically focus on MANO fitting using multi-view images.

The process begins with estimating 2D key points for each view using AlphaPose [20]. These 2D keypoints are triangulated using the intrinsic and extrinsic camera parameters associated with each view to obtain 3D keypoints. The MANO [2] model inputs pose and shape parameters and outputs the corresponding hand mesh along with joint locations. The fitting process aims to update these pose and shape parameters to minimize the difference between the model's predicted and triangulated 3D key points. This minimization is typically achieved by calculating the mean squared error (MSE) as the loss function between the corresponding 3D key points. The parameters are then optimized using a gradient descent algorithm.

For further refinement, if a coarse mesh of the hand is available (for instance, generated using InstantNGP [21]), the point-to-surface loss can be utilized to update the shape parameters with higher accuracy. This additional step allows the model to capture the finer details of the hand's shape and improves the overall accuracy of the pose estimation.

2.3.2 Inverse Kinematics

To obtain the joint angles of the hand and its global orientation, an optimization-based approach inspired by [22] is used. Specifically, the joint angles, global rotation, and global translation are treated

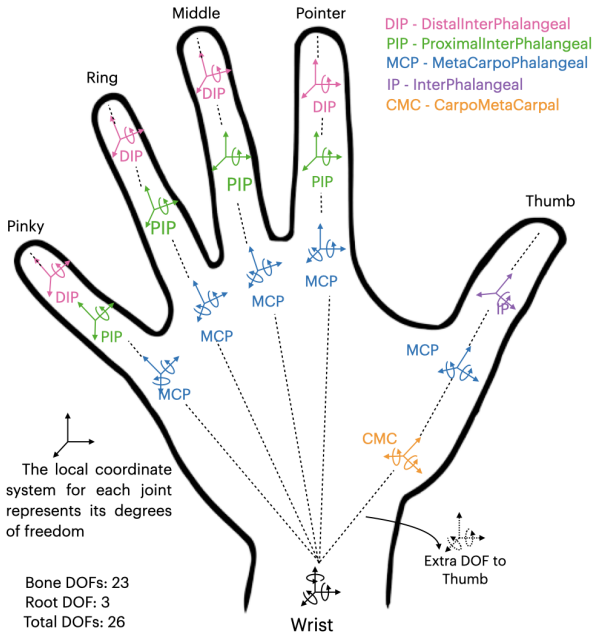

Figure 2.3: Figure showing the degrees of freedom of rotation for each of the joints.

as optimization parameters Θ. Then a forward kinematics ($Fk(\Theta)$) pass is performed which takes the joint angles as input and outputs 3D joint locations. As the forward pass is differentiable, gradient descent is used to obtain the optimal parameters that explain the given 3D joint positions. Finally, the L2 loss is minimized between predicted and target key points:

$$\mathcal{L}_{kyp} = ||Fk(\Theta) - x||^2 \tag{2.16}$$

where x are the 3D joint locations predicted by AlphaPose [20].

To avoid the invalid hand poses, anatomical constraints (See Figure 2.3) and joint angle limits are applied by applying a hinge loss as limit loss $\mathcal{L}_{lim}$ as follows:

$$\mathcal{L}_{lim} = \sum_{i=1}^{|\Theta|}((\max(0, ||\Theta^i - l_h^i||^2) + \max(0, ||l_l^i - \Theta^i||^2)) \tag{2.17}$$

where l_l and l_h are the lower and upper limits on joint angles, respectively.

The final loss function is given by:

$$\mathcal{L} = \mathcal{L}_{kyp} + \lambda\mathcal{L}_{lim} \tag{2.18}$$

The Adam [23] optimizer is used with a learning of 0.001 to optimize the loss function. The value of λ is set to be 1. The current frame is initialized based upon previous frame, this helps in faster convergence and helps in maintaining temporal consistency.

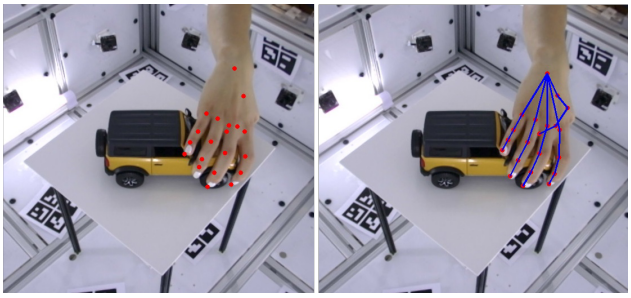

Figure 2.4: The left figure shows the backprojected 3D keypoints predicted by AlphaPose [20]. The right figure shows the fitted hand skeleton using inverse kinematics.

Once joint angles are obtained, one euro filter [24] is applied to the joint angles to smoothen any high-frequency jitter in the sequence.

Chapter 3

RealGrasper: Learning Human Hand Grasping from Multi-View Images

As discussed in previous chapters, exploring neural field representations is crucial for addressing the problem of novel grasp generation using multi-view image cues. This requires developing new methods to represent hand-object interactions and creating a multi-view image dataset to observe these interactions.

In this chapter, we demonstrate how tackling the challenges of dataset creation and representation enables us to learn a model of human grasping directly from real-world multi-view images. First, we introduce RealGrasp, a 53-view RGB dataset with over 362K frames spanning 11 different objects, 4 subjects, 17 free-hand sequences, and 20 grasps per object per subject. Next, we show how this dataset can help build high-fidelity template-free neural models of hands, objects, and grasps with minimal supervision. Rather than use mesh-like representations that may not faithfully capture appearance and contact properties, our neural models are neural fields that model shape, appearance, and even contact regions from arbitrary viewpoints. Finally, we introduce RealGrasper, a generative model consisting of a conditional variational autoencoder that estimates the final grasp pose of the hand when given an initial 3D hand pose and object shape. We show quantitative and qualitative results to evaluate our dataset and representation of grasping model.

3.1 Introduction

Each day, as we go about our daily lives, we effortlessly grasp more than a hundred different objects [25] thousands of times [26]. Grasping, a task so ordinary for humans, remains tremendously difficult for machines as evidenced by its extensive study in robotics [27] and computer vision [28]. Understanding human grasping has important applications for instance in robotics, mixed reality, and activity recognition. However, progress has been limited by challenges in capturing rich real-world grasping data and building suitable representations to capture hands, objects, and grasps.

Because of these challenges, previous work has resorted to simulation [7, 29–32] as a way to model human grasps. In simulation, physical laws and heuristics are used to model the components of grasping including contact forces, friction, mass, and gravity. Inevitably, modeling every source of physical variation is difficult resulting in a domain gap between simulation and the real world [33]. Some meth-

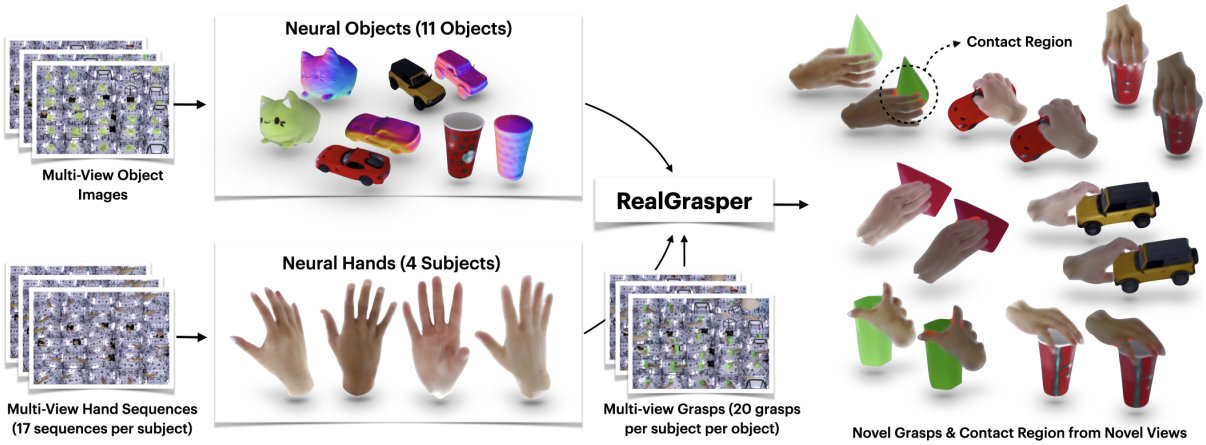

Figure 3.1: We present a method to learn a model of human hand grasping directly from real-world multi-view images. First, we introduce RealGrasp, a large 53-view RGB dataset with over 362K frames spanning 12 different objects, 4 subjects, 17 free-hand sequences, and 20 grasps per object per subject. We introduce neural field representations of objects and hands that capture shape and appearance (middle). We show how our dataset and representation can be used to learn a generative grasp model, RealGrasper, that estimates the final hand pose of the grasp when given an object and the initial hand pose. RealGrasper generalizes to previously unseen objects and can visualize shape, appearance, and even contact regions (rightmost, denoted by red regions).

ods combine known physical constraints (e.g., contact) with real-world observations but: methods that use observations from markers can hinder free hand motion [10, 34–36] while methods that operate on images use representations like parametric hand models [37, 38] that lack the expressive capability to easily model 2D hand boundaries, surface, and contacts. Furthermore, existing real-world grasping datasets [3, 12] have been limited to providing coarse 3D hand pose, use specially designed or instrumented objects, and do not enable modeling of both the **appearance and geometry** of grasps.

We show that addressing the dataset and representation challenges can enable us to learn a model of human hand grasping directly from real-world multi-view images. To this end, we present **RealGrasp**, a new 53-view RGB dataset with over **362K** image frames: multiple views of **11 different objects**, 17 multi-view videos of free hand articulation across **4 subjects**, and multiple views of **20 different grasps** on each of the objects and subjects – all captured without any special markers or sensors. We use this dataset to build high-fidelity neural hand and object models with minimal supervision (only 3D camera poses and hand poses obtained from off-the-shelf methods). Rather than use meshes as representations of the hand and object, we build on the latest advances in neural fields, specifically neural shape [39] and articulating radiance fields [17, 18]. Our **template-free** high-fidelity neural hand and object models learn appearance, geometry, and can model the **contact regions** of the grasp.

The neural hand and object models are then used to learn **RealGrasper**, a generative model consisting of a conditional variational autoencoder (CVAE) that, when given an initial 3D hand pose and an object model, estimates a final hand pose representing a plausible grasp of the object. During training, RealGrasper is only trained on multi-view RGB images and derived 3D hand poses of grasps without

any other supervision. Our method produces novel natural grasps and can visualize the appearance and geometry of grasps from arbitrary viewpoints as shown in Section 4.1. We quantitatively evaluate our dataset and representation, and justify key design choices in Section 4.5. To our knowledge, ours is the first method to use neural fields to model grasping on real data making comparison with other methods challenging – but we provide some comparisons [11]. To sum up our contributions:

- **RealGrasp**, a large real-world 53-view RGB dataset (which we will release publicly) with over **362K** frames captured across 11 objects, 4 subjects, 17 free-hand sequences, and 20 grasps per object per subject.
- We model shape, appearance, and contact of hand–object grasping with **neural representations** faithfully to images.
- **RealGrasper**, a generative CVAE that learns to synthesize photorealistic grasping given an object and initial 3D hand pose from multi-view images.

3.2 Related Work

In this review of related work, we focus on datasets, grasp simulation, perception of hand-object interactions, and representations.

Datasets: Datasets for human grasps are challenging to obtain because they need specialized hardware, extensive human annotation, and significant post-processing to make them useful. Some datasets use markers or special gloves to track the hand and object [34, 36, 40, 41] but this hinders natural hand motion and introduces changes in image appearance. Therefore, work has focused on manual annotations [42–45], optimization [46], or automatic annotation [47] from RGB or depth. Many of these datasets are limited to only 3D hand poses and lack information about hand surface and contacts. Synthetic datasets [37, 48, 49] suffer from a domain gap that makes it challenging to generalize to real data. Other datasets like InterHand2.6M [50, 51] are limited to hand only without any objects, while others [52] focus on 2D understanding only.

ContactDB [12] and ContactPose [53] aim to address these limitations, and focus on scaling to many users and objects. While ContactDB is captured using thermal imaging, ContactPose uses multi-view RGB-D data. Both methods are limited to 3D hand poses only, objects are not real, and do not have sufficient views to support neural field representations. In this chapter, we focus on providing a high-quality dataset with sufficient views to support neural field representations, enable capture of both appearance and geometry, and enable grasping models (including contact) to be trained from multi-view images.

Simulation for Grasping: Due to challenges in capturing real data, there has been extensive work on using simulation for modeling human grasps. GraspIt! [7] is one of the most widely used methods and uses hand-designed heuristics and physics to obtain a final hand grasp pose given an object and initial hand pose. More recent multi-finger grasp simulation methods rely on analytic methods [29–

31, 54] and can be used with a human hand model. Recently, D-Grasp [8] introduced a reinforcement learning method for dynamics grasp syenthesis, and Grasp'D [32] introduced differentiable simulation for grasping. ManipHand [10] combines marker-based motion capture with a learning-based approach to synthesize manipulations of objects. All of the above methods suffer from a domain gap to real data [33].

Perception for Grasping: Simultaneously in computer vision, significant work has studied capturing hands interacting with objects [11, 37, 43, 55–59]. Several methods combine perception with physical constraints (proximity, contact, forces) during optimization or learning [35, 42, 60–62]. To make hand shape and pose estimation easier, parametric hand models have been developed, notably MANO [2] and Total Capture [63], which are used by several methods for pose estimation [38]. However, parametric/template hand models cannot capture hand boundaries well resulting in a mismatch between shape and appearance. On the contrary, we propose to use template-free methods for obtaining hand and object models with better shape–appearance alignment.

Representations: Recent advancements in coordinate-based neural networks, or neural fields [1], have shown great success in encoding the geometry [64–66] and appearance [17, 67, 68]. For example, neural radiance field (NeRF) [17] uses an MLP to model the density and color and achieves photorealistic novel view synthesis. VolSDF [39] and NeuS [69] improve the geometry representation and reconstruction of NeRF by deriving the density from a signed distance function (SDF) representing the distance to the closest surface of the scene geometry. Instant-NGP [21], Plenoxels [70], and TensoRF [71] greatly reduce the cost of building NeRF models. Many approaches also explore articulated neural fields to model dynamic human body [18, 72–75]. LISA [76] proposes an implicit hand model, but code and datasets are not publicly available. TAVA [18] proposes a template-free animatable neural representation for dynamic actors (e.g., human bodies), which is robust to unseen poses. We show how neural field representations, specifically TAVA [18] and VolSDF [39], can be used to build a neural representation of grasps from real data.

3.3 RealGrasp Dataset

We first describe our RealGrasp dataset, in particular, the hardware capture system, capture protocol, and annotation. The RealGrasp dataset was driven by three key considerations: (1) capture hands interacting with objects without any markers or special sensors like depth or thermal cameras, (2) capture both the appearance and geometry of grasps, and (3) support neural shape and radiance fields as representations for learning grasping. Achieving this goal from purely RGB videos requires a multi-view capture system with known camera poses. Many prior datasets (see Section 3.2) contain multi-view images or video of hand grasps [34, 46, 47], but none have the large number of views needed to support neural field representations or are limited to hands only [50]. Thus, we built a custom system to capture a large 53-view real-world dataset of hand grasps.

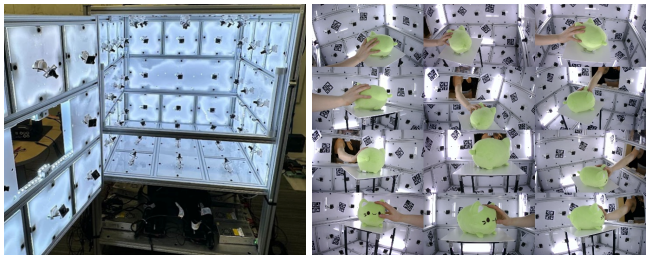

Figure 3.2: (Left) Our data capture system where hands, objects, and grasps are captured by 53 cameras. (Right) Sample grasp frame from 12 out of the 53 views in our dataset.

Data Capture System: The data capture setup is shown in Figure 3.2 (left). It consists of 53 RGB cameras uniformly located inside a cubical capture volume with each cube face consisting of 9 cameras. The sides of the cube are illuminated evenly using LED lights with additional edge lights. Each RGB camera records at 120 FPS with a resolution of 1280×720. This system capture both static (for objects) and dynamic scenes (for hands and grasps). The cameras are software synchronized with a frame misalignment of no more than 3 ms. The multi-view system is calibrated for camera intrinsics and extrinsics using COLMAP [77, 78] with fiducial markers on the walls.

RealGrasp Dataset: RealGrasp is a large real-world multi-view RGB dataset of hands grasping natural objects that we will publicly release. It contains **362K** image frames: multiple views of **11 different objects**, 17 multi-view video sequences of free hand articulation with **4 subjects**, and multiple views of **20 different grasps** on each of the 11 objects for all 4 subjects. Of the total frames, we use 360K to create neural hands, 636 frames for neural objects, and 1920 frames for grasp learning. Our goal is not to compete with existing datasets on quantity, but instead we focus on enabling the use of neural field representations for grasps. Figure 3.2 (right) shows some example data from our dataset.

Data Capture Protocol: Our capture protocol consists of four steps. First, we capture a sequence of an empty scene for camera calibration and background subtraction. Next, we collect multi-view videos of hands to build neural hand models (see Section 3.4.1), subjects reach their right hand into the center of the box and move their hand in different motions. Then, we collect static multi-view images of objects for neural object models (see Section 3.4.1). Finally, we record multi-view images of our subject's hand grasping the object in 20 different ways.

Automatic Annotation: The appearance and geometry of hand grasps are automatically extracted by our method as described in Section 4.4. Apart from this, we also provide 2D and 3D hand joint locations which we obtain from OpenPose [47] followed by 3D triangulation over the multi-view hand sequence. Then, we use inverse kinematics optimization [22] to obtain the joint angles and global orientation of the hand skeleton by optimizing them using gradient descent. We impose constraints to limit the degrees of freedom and joint angles for the rotation of the bones as described in Figure 2 in the supplement. To achieve temporal smoothness for the sequence, we apply the 1€ Filter [24] on the estimated parameters.

To segment the hand and object from the background, we use a combination of both traditional and learning-based background subtraction methods [79, 80]. However, segmenting objects using the above

methods fails in many cases due to complex object texture, so we use PhotoRoom [81], a commercial application. To ensure segmentations are consistent across views, we first train InstantNGP [21] and extract an alpha mask.

3.4 Method

We aim to accurately capture hands, objects, and grasps from real-world observations with the ultimate goal of generating human grasps that are natural and realistic in terms of appearance, shape, and physical contact. Prior work in grasping has struggled to faithfully capture the appearance of hand-object interaction due to the limitations of commonly used mesh representations. We address this issue by leveraging neural shape and appearance fields to represent visual details of hands and objects as described in Section 3.4.1. Our neural representations are learned with only minimal supervision for 3D hand and camera poses obtained using off-the-shelf methods [47, 77].

In Section 3.4.2, we introduce our generative model RealGrasper, a conditional variational autoencoder (CVAE) [82] built on our neural representations to synthesize natural hand grasps given the encoded shape knowledge of the target object and initial hand pose. RealGrasper is trained solely on multi-view videos of hands grasping objects without any other supervision to generate high-quality photorealistic renderings of human grasps. In Section 3.4.3, we explain how the losses used to train RealGrasper.

3.4.1 Neural Grasp Representation

Prior works in hand-object interaction (see Section 3.2) heavily rely on mesh representations of object or parametric models such as MANO [2]. Due to the low dimensional nature of these template meshes, they can result in misalignment when fit to images, which adversely affects the estimation of hand-object contact and prevents the extensive study of real-world human grasping. Thus, it is important to use a representation that can reconstruct the appearance and geometry of hands and objects faithfully and avoid image misalignments. Inspired by the recent success of neural fields, we build an object representation upon VolSDF [39] and a hand representation based on TAVA [18] to learn a neural representation of hand grasps. These representations are derived from Neural Radiance Fields (NeRF) [17] which encodes the geometry and view-dependent appearance of a scene as a continuous field of radiance $\mathbf{c}(\mathbf{x}, \mathbf{v})$ and volume density $d(\mathbf{x})$ using a multi-layer perceptron (MLP) $f : (\mathbf{x}, \mathbf{v}) \rightarrow (\mathbf{c}, d)$ where $\mathbf{x} \in \mathbb{R}^3$ is a 3D point and $\mathbf{v} \in \mathbb{R}^3$ is the corresponding viewing direction. The radiance field along each camera ray $\mathbf{r}$ is given as:

$$\mathbf{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - exp(-d_i\delta_i))\mathbf{c}_i, \text{where } T_i = exp(-\sum_{j=1}^{i-1} d_i\delta_i), \tag{3.1}$$

δ_i is the distance between adjacent sample points on the camera ray and T_i denotes transmittance.

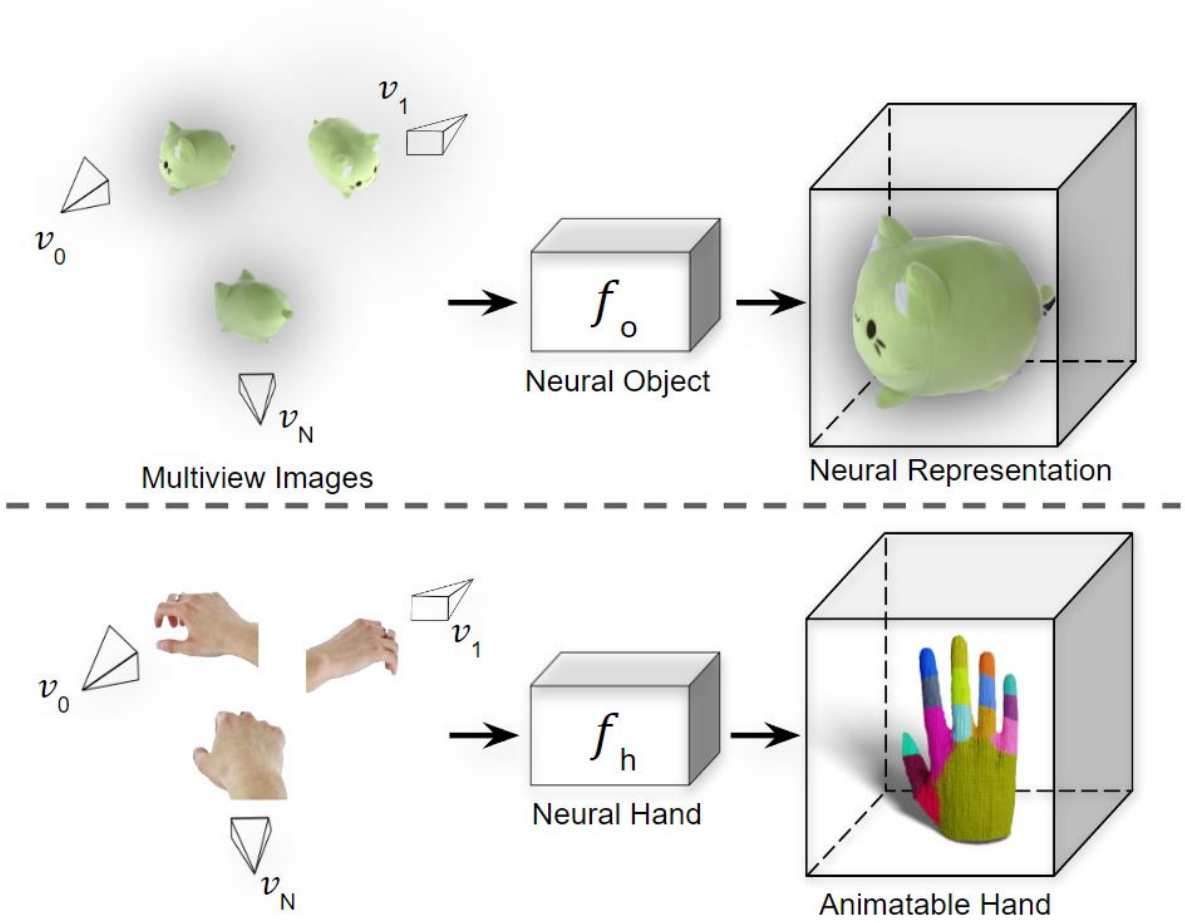

Figure 3.3: To learn a neural representation of hand-object interaction, we train a neural radiance field of static objects using VolSDF [39] and a dynamic neural hand model using TAVA [18] from RealGrasp dataset captured by our multiview camera system.

Neural Object Representation: To accurately represent the shape and appearance of objects, we train a VolSDF [39] model from multi-view images. VolSDF is a neural field that models the volume density d as the Laplace's cumulative distribution function Ψ of a learnable signed distance function (SDF): $d(\mathbf{x}) = k\Psi(-SDF(\mathbf{x}))$. The zero-level set of the SDF defines the shape of object's surface. Such a formulation of density improves the geometry reconstruction compared with vanilla NeRF representation. To render the VolSDF model of the object, we follow Equation (3.1), but the radiance $\mathbf{c}(\mathbf{x}, \mathbf{v}, \mathbf{n})$ is also dependent on the level set's normal $\mathbf{n}(\mathbf{x}) = \nabla_{\mathbf{x}} SDF(\mathbf{x})$. This representation allows us to synthesize realistic novel views as well as reconstruct the object shape of the object with high fidelity and use it for grasp generation.

Neural Hand Representation: Unlike static objects, hands exhibit complex articulated poses and neural representation adopted for objects lacks the capacity to model dynamic, deformable scenes and articulated actors. To learn a neural representation of the hand model, we need the ability to animate the hand and generalize to out-of-distribution poses unseen during training. Hence, we adopt TAVA [18], a

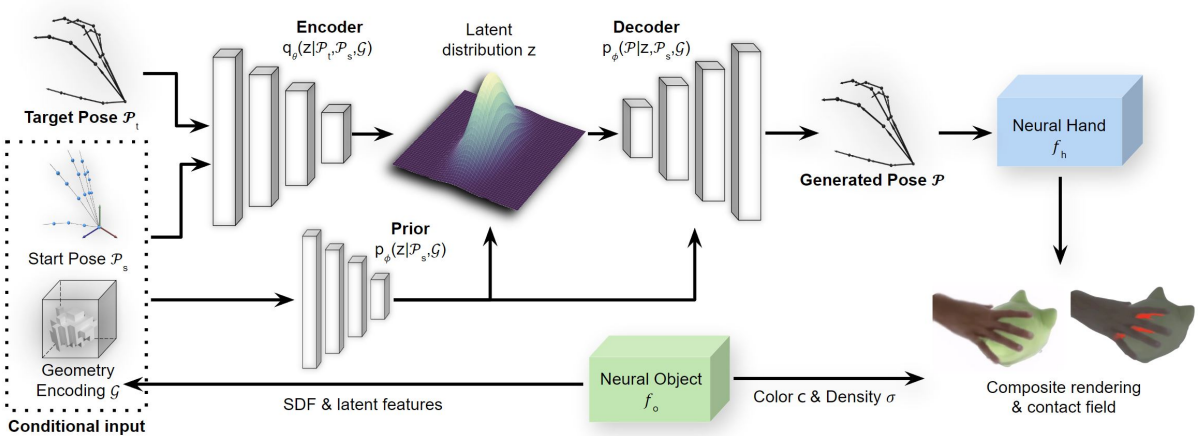

Figure 3.4: **RealGrasper architecture.** We build on top of the CVAE model introduced in [83]. Given the start pose $\mathcal{P}_\mathbf{s}$, geometry encoding $\mathcal{G}$ of the target object from Neural Object model, and the ground truth target pose $\mathcal{P}_\mathbf{t}$, the decoder reconstructs the hand pose by sampling from the estimated posterior from encoder during training. At inference, we can sample from the learned prior and generate novel grasps from decoder given the conditional input $\mathcal{P}_\mathbf{s}$ and $\mathcal{G}$. The generated poses $\mathcal{P}$ can then be used to synthesize novel views of realistic hand grasps and contact regions (red).

template-free animatable neural radiance field that allows us to drive the hand model given novel poses at test time. The Neural Hand representation consists of a Lambertian neural radiance field that represents the shape and appearance of hands, and a neural blend skinning function to animate hands. The neural radiance field adopts Mip-NeRF [84]. The neural skinning function predicts skinning weights at each 3D points to blend all bone transformations using forward LBS-based deformation. We can render the deformed hand using Equation (3.1) after finding the radiance $\mathbf{c}(\mathbf{x}, \mathbf{v})$ and density $d(\mathbf{x})$ of the sampled points in their canonical space via inverse skinning. The Neural Hand model are trained solely on multi-view images and 3D hand poses obtained from off-the-shelf methods [47].

Composite Grasp Representation: To model grasps, we combine the Neural Object and Neural Hand models in a photorealistic way. To combine the two neural fields, we use an additive composition of Equation (3.1) same as in [85]:

$$\begin{aligned} \mathbf{C}(\mathbf{r}) &= \sum_{i=1}^{N} T_i((1 - exp(-d_i^o \delta_i))\mathbf{c}_i^o + (1 - exp(-d_i^h \delta_i))\mathbf{c}_i^h) \\ T_i &= exp(-\sum_{j=1}^{i-1}(d_i^o \delta_i + d_i^h \delta_i)), \end{aligned} \tag{3.2}$$

where $\mathbf{c}_i^o$, d_i^o denotes radiance and density of the object and $\mathbf{c}_i^h$, d_i^h denotes radiance and density of the hand. In the end, we can synthesize a photorealistic rendering of the hand grasping the object given any camera viewpoint.

Contact Region Reasoning: The neural field representation also allows us to extract the contact field between the object and the hand. Intuitively, contact is likely to occur in the vicinity of the object surface where the hand volume density is high. Thus, we query the volume density of Neural Hand at the sampled points close to the zero level set of the object SDF. If the density of a part of the hand is above a high threshold at regions in close proximity to the surface of the object, we set a positive mask for those parts of the hand and the contact field can be visualized using the mask which is visualized in red in Figure 3.5.

3.4.2 Neural Grasp Generation

The goal of RealGrasper is to synthesize novel grasps for target objects. By utilizing a generative model that learns to reconstruct a target hand pose from multi-view images of hands grasps given the target object and starting hand pose, RealGrasper is able to produce plausible grasps without any additional supervision. Built on Neural Grasp representation introduced in Section 3.4.1, our model can synthesize high-quality appearance and geometry of novel grasps from arbitrary viewpoints and model the contact between the hand and the object. Inspired by [83], we adopt a conditional variational autoencoder (CVAE) [82] architecture which formulates grasp reconstruction $p_\phi(\mathcal{P}_t|\mathcal{P}_\mathbf{s}, \mathcal{G})$ as a latent variable model as shown in Figure 3.4.

Input Parameterization: We take the 21-joint hand skeleton in OpenPose [47] and parameterize the hand pose $\mathcal{P} = [\mathbf{t}\ \mathbf{r}\ \mathcal{J}\ \mathbf{\Phi}]$ as the translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\mathbf{r} \in \mathbb{R}^3$ of the root joint together with joint positions $\mathcal{J} \in \mathbb{R}^{20\times3}$ and bone rotation $\mathbf{\Phi} \in \mathbb{R}^{23}$ for the rest of the joints relative to the root joint. Since the hand skeleton is subject to a kinematic structure and a certain range of configurations, we can limit the degrees of freedom for the rotation of bones (see supplementary). The object geometry encoding $\mathcal{G} = [SDF(\mathcal{J})\ \mathcal{S}]$ consists of the SDF queried at the joint locations $SDF(\mathcal{J}) \in \mathbb{R}^{20}$, and shape features $\mathcal{S} \in \mathbb{R}^{32}$ on the object mesh extracted from a pretrained encoder [86].

Conditional Prior: We first learn a conditional prior from which the latent variables $\mathbf{z} \in \mathbb{R}^{24}$ represent the possible grasping motion transitions from the starting pose to the target pose on the object:

$$p_\phi(\mathbf{z}|\mathcal{P}_\mathbf{s}, \mathcal{G}) = \mathcal{N}(\mathbf{z}; \mu_\phi, \sigma_\phi),$$

which parameterizes a Normal distribution with mean μ_ϕ and variance μ_ϕ. estimated by an MLP. Intuitively, the distribution of possible grasping motion could vary given different starting poses and objects. Thus, explicitly learning the prior helps the CVAE to generalize to diverse grasps and stabilize the training in our experiments.

Encoder and Decoder: The encoder learns the approximate posterior for training and parameterizes a Gaussian distribution

$$q_\theta(\mathbf{z}|\mathcal{P}_\mathbf{t}, \mathcal{P}_\mathbf{s}, \mathcal{G}) = \mathcal{N}(\mathbf{z}; \mu_\theta, \sigma_\theta).$$

We use KL divergence loss to regularizes the posterior to be near the prior (see Section 3.4.3). Conditioned on the latent variable $\mathbf{z}$ sampled from the encoder posterior, the starting hand pose and object

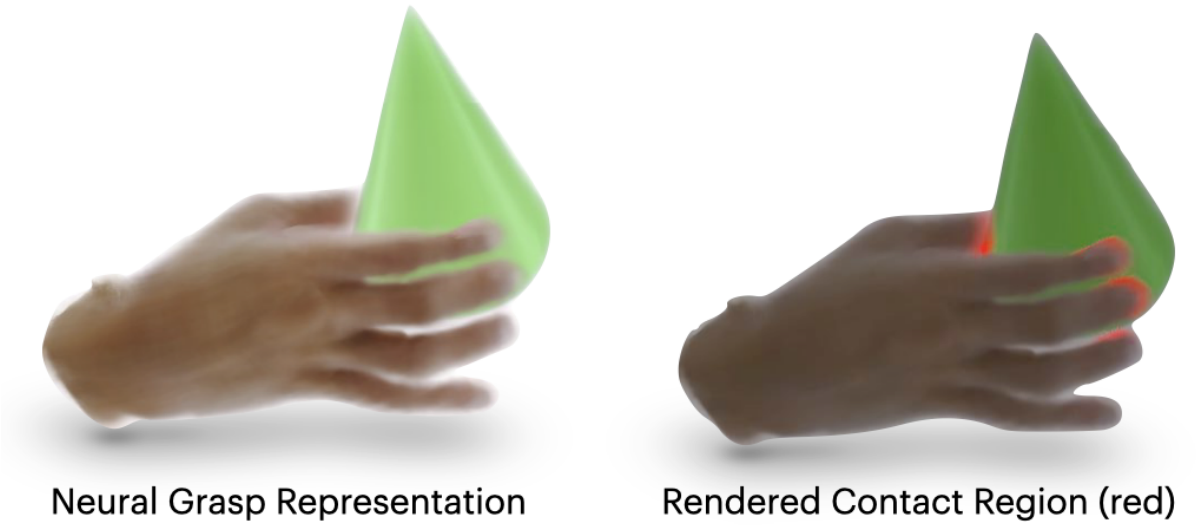

Figure 3.5: **Contact Regions**. The red highlighted part of the fingers grasping the cone indicates close contact where the signed distance of the hand volume to the zero level set of the object is smaller than a threshold.

geometry $\{\mathcal{P}_s, \mathcal{G}\}$, the decoder reconstructs the target pose $\mathcal{P}_t$ during training thereby learning the likelihood $p_\phi(\mathcal{P}_t|z, \mathcal{P}_s, \mathcal{G})$.

Generating Novel Grasps: At inference, the decoder takes the concatenation of the conditional input $\{\mathcal{P}_s, \mathcal{G}\}$ and sampled $\mathbf{z}$ from the learned prior $p_\phi(\mathbf{z}|\mathcal{P}_\mathbf{s}, \mathcal{G})$ to generate novel grasping poses $\mathcal{P}$. The generated grasping hand pose can be applied to synthesize photorealistic hand using our trained Neural Hand model. Since Neural Object and Neural Hand use radiance field representation, we can jointly render them using volumetric rendering in Equation (3.2) (see Section 3.4.1). During training, we can optimize the rendering loss between the synthesized composite hand-object image and the captured image, as explained in the next section.

3.4.3 Model Training and Loss

We first train Neural Object and Neural Hand following the original training setup in VolSDF [39] and TAVA [18] using multi-view images of objects and hands in our RealGrasp dataset. We then train RealGrasper with multi-view grasp images and hand poses to reconstruct grasp poses given initial hand pose and object encodings. The variational lower bound of the CVAE [82] is optimized while Neural Object and Neural Hand are fixed:

$$\begin{aligned} \log p_\phi(\mathcal{P}_t|\mathcal{P}_s, \mathcal{G}) \geq \mathrm{E}[\log p_\phi(\mathcal{P}_t|z, \mathcal{P}_s, \mathcal{G})] \\ -KL(q_\theta(\mathbf{z}|\mathcal{P}_\mathbf{t}, \mathcal{P}_\mathbf{s}, \mathcal{G})\|p_\phi(\mathbf{z}|\mathcal{P}_\mathbf{s}, \mathcal{G})), \end{aligned} \tag{3.3}$$

	Hands			Objects										
	subject1	subject2	subject3	cat	dog	couch	cup	table	car1	car2	car3	pyramid	prism	cone
PSNR↑	23.81	25.04	24.38	35.93	32.71	23.76	20.81	29.92	24.65	23.19	23.57	37.21	37.68	37.25
SSIM↑	0.87	0.81	0.78	0.95	0.94	0.79	0.82	0.91	0.84	0.87	0.87	0.97	0.98	0.97

Table 3.1: We report PSNR and SSIM as a measure of visual appearance quality of our representations Neural Hand and Neural Object on all subjects of hands and objects in our RealGrasp dataset. The higher the score the better the image quality.

where the first term measures the reconstruction error $L_{\mathcal{J}}$ of the decoder and the KL divergence $L_{\mathcal{KL}}$ regularize the posterior distribution approximated by the encoder to be close to the prior. In addition, we impose a rendering loss L_{rgb} on the composited image of the generated hand grasp, and a regularization loss to encourage physically plausible contact $L_{contact}$. In summary, our training loss consists of four terms:

$$L = L_{\mathcal{J}} + \alpha L_{\mathcal{KL}} + \beta L_{rgb} + \lambda L_{contact}, \tag{3.4}$$

where α, β, λ are hyperparameters to balance the loss terms. The primary objective of the model during training is to minimize the reconstruction error between the joint locations of the target hand $\mathcal{J}_t$ and the generated hand $\hat{\mathcal{J}}$:

$$L_{\mathcal{J}} = \|\mathcal{J}_t - \hat{\mathcal{J}}\|^2. \tag{3.5}$$

We encourage the posterior distribution to be close to the estimated prior distribution by minimizing the KL divergence:

$$L_{\mathcal{KL}} = KL(\mathcal{N}(\mathbf{z}; \mu_\theta, \sigma_\theta) \| \mathcal{N}(\mathbf{z}; \mu_\phi, \sigma_\phi)). \tag{3.6}$$

The volumetric rendering loss L_{rgb} between the final synthesized image and the ground truth image is similar to the photometric loss in NeRF [17]:

$$L_{rgb} = \sum_{\mathbf{r}} \|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\|. \tag{3.7}$$

To avoid penetrating the object, we add a soft regularization to penalize negative distance to the object surface at hand joints:

$$L_{contact} = -\sum_{\mathbf{x} \in \mathcal{J}} (1 + \exp^{-k \cdot SDF(\mathbf{x})})^{-1}, \tag{3.8}$$

where k is a hyperparameter set to 20.

3.5 Experiments & Results

In this section, we show results and evaluate various components of our method including the quality of our dataset, representation, grasping, and ablate on design choices.

Figure 3.6: **Contact Loss Ablation**. The left image shows the generated grasp when no contact loss is used. The right image shows a more plausible grasp generated when applying the contact loss demonstrating the effectiveness of the contact loss as a regularizer to discourage penetration.

	Training MPJPE↓			Test MPJPE↓	
	car2	cup	cone	car1	prism
w/ shape encodings	**2.23**	**1.98**	**0.55**	**4.53**	**3.64**
w/o shape encodings	3.20	4.85	2.56	4.95	4.52

Table 3.2: We show that the MPJPE errors of grasp pose reconstruction without shape encodings is consistently greater on both training and test objects. This indicates that shape encoding is critical to our model. We multiplied the error numbers by 1000 to adjust the scale.

Implementation Details: Our model is implemented in PyTorch Lightning. The CVAE of RealGrasper is implemented using MLPs with 8 layers and Leaky ReLU as activation. We use ADAM optimizer with a learning rate of $5e^{-4}$ and batch size of 1 with gradient accumulation of 8 batches for training on a single RTX 2080Ti GPU. We train Neural Hand for each subject on four Tesla V100 for 72 hours and Neural Object on one V100 for 16 hours. We set the loss weights $\alpha = 1.0, \beta = 1.0, \lambda = 1.0$. To train the Neural Grasp, we consider 9 objects for training and 2 objects (prism and car1) for test split out of total 11 objects.

Dataset Quality: The hand pose estimation using OpenPose sometimes yields incorrect or missing keypoints for certain frames. We filter invalid frames by checking if the tracked hand skeleton is complete. After filtering, 95% of the frames in our dataset are reliable. Finally, we perform histogram equalization to improve the image contrast.

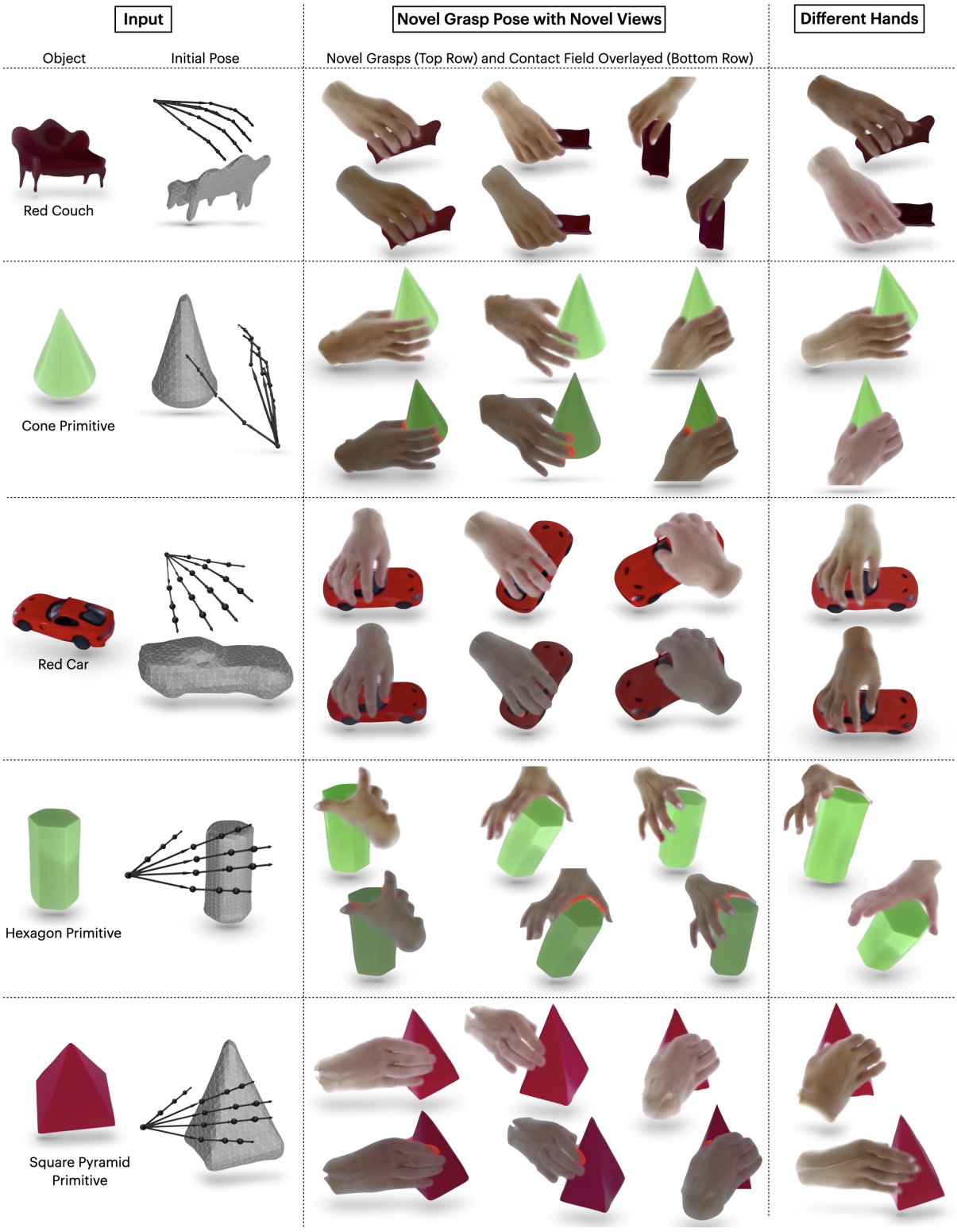

Figure 3.7: Visualization of generated grasps and contact from RealGrasper given input object and initial hand pose shown in different views. The right column shows grasp synthesis on same input using Neural Hand model from different hand subject, which does not appear during training. The results demonstrate our grasp generation model can generalize to different hand subjects.

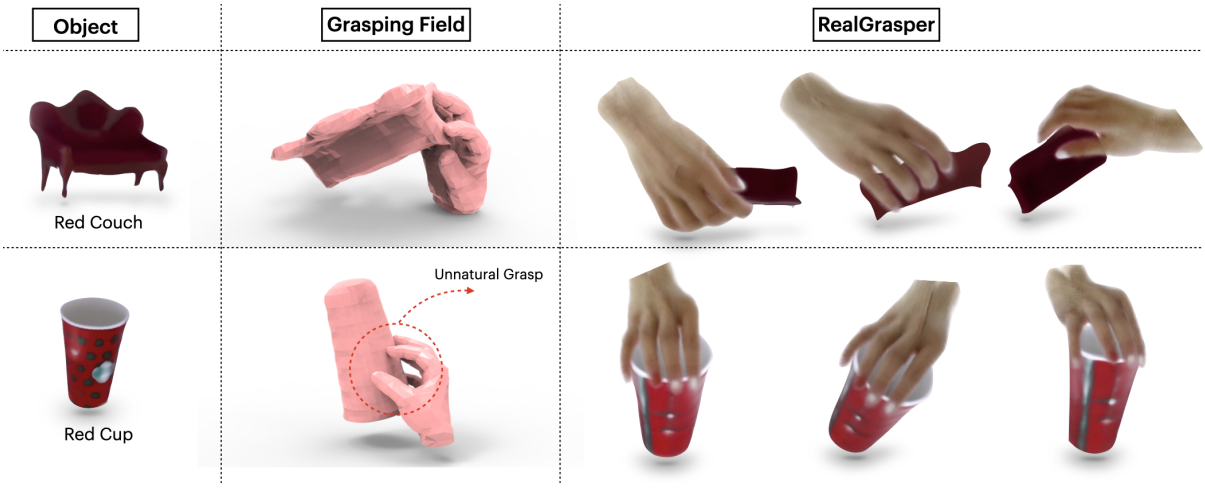

Figure 3.8: Qualitative comparison with GraspingFields [11]. Our RealGrasper enables photo realistic rendering of hand-object grasping. We demonstrates more human-like natural grasps of the couch model and cup compared with Grasping Field and visually appealing rendering quality.

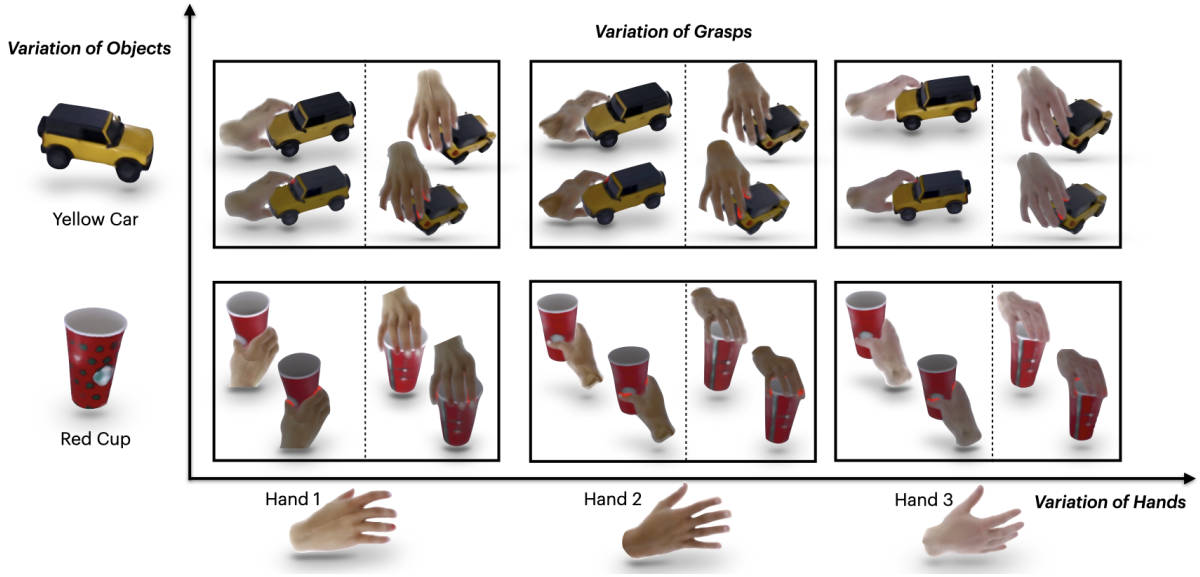

Figure 3.9: **Variation of Hands, Objects and Grasps**: Our framework learns hand representations from different subjects incorporating subject's unique characteristics. We can swap hands and objects to generate various grasps in a plug and play manner. Hence, for a certain object, we can show variety of grasps with different hands.

3.5.1 Representation Evaluation

In this section, we evaluate the quality of our neural field representations of the hand, object and grasps.

Quantitative Evaluation: We measure the visual quality of our neural representation using PSNR and SSIM metrics (higher is better) on our RealGrasp dataset (see Table 3.1). For Neural Object and

Neural Hand, we render five novel views and report the average metric value for all 3 hand subjects and 11 objects. Our PSNR quality is consistently over 25, with a few objects yielding lower PSNR/SSIM scores due to theur small size. To our knowledge, ours is the first neural field-based representation for grasping, thus there are no other methods we can compare against.

Qualitative Evaluation: We perform qualitative evaluation and visualize renderings of the novel grasp poses with varying novel views across multiple objects and associated contact field as well as rendering of same object grasps with different neural hand representations learnt on different individuals, as shown in Figure 3.7. Our proposed representation is able to produce realistic novel views of grasps for various combinations of hands and objects.

3.5.2 Comparison to Previous Work

To our knowledge, we are first method to model grasping using neural fields from real multi-view image data. However, this makes it challenging to directly compare with previous work. Works such as Grasp'D [9] and D-Grasp [8] are based upon physics simulation while others use parametric models [35, 38]. Furthermore, these methods work by taking object's geometry as their input and do not model appearance. On the other hand, our method can be trained and evaluated by considering only images as input. We therefore provide qualitative comparisons with GraspingFields [11] to show the difference in the quality of grasps and renderings.

Qualitative Comparisons: We show qualitative results with Grasping Field on two objects car and couch as shown in Figure 3.8. Both of the methods are able to grasp the object similarly, but our method produces better contact compared to Grasping Fields. Additionally, our method models appearance, can generate superior photo-realistic composite rendering of the grasp from arbitrary viewpoints, and implicitly extracts contact regions.

3.5.3 Ablation Study

We ablate on different components of our method, in particular, contact loss and the need for shape encodings.

Effect of Contact Loss: We perform an ablation on the impact of contact loss on the performance of our results. Contact loss acts as a regularizer in the total loss term which penalizes the network to make predictions inside the object mesh. Specifically, contact loss acts as a soft regularizer if the SDF of the object at the joint bone locations is negative. We show in Figure 3.6 that the contact loss improves the physical contact between hand and object.

Effect of Shape Encodings: To condition our CVAE on the shape information about the object shape, we use [86] to encode the shape. This allows us to integrate local information and incorporate translational equivariance in the form of shape encodings. To test our hypothesis, we do two experiments one with shape encodings and another without shape encodings and report Mean Per Joint Position Error (MPJPE) on both training and test objects as shown in Table 3.2. The MPJPE measures Euclidean error

averaged over all hand joints. Results show that shape encoding is essential to provide knowledge of the object shape to help training the CVAE.

3.6 Conclusion

In this chapter, we addressed the dataset and representational challenges in understanding human grasping from multi-view video. We introduced RealGrasp, a large frame multi-view RGB dataset designed to support neural field representations to model grasping. It consists of over **362K** frames spanning 11 different objects, 4 subjects, 17 free-hand sequences, and 20 grasps per object per subject. We then showed how this data can support the representation of hands, objects, and grasps as neural fields. Finally, we use the neural representations to train RealGrasper, a grasping model that generates plausible grasps given an object and initial hand pose.

Limitations and Future Work: Our approach has several limitations. First, our dataset is currently limited in the number of objects/subjects and only static grasps – we plan to extend this work to dynamics grasps and in-hand manipulation. Our neural models take a significant amount of time to train and generate composites which we hope to address by investigating faster neural fields [21, 70]. Despite designed loss functions to avoid collision, our model could still fail at inference when the hand and object penetrate. We consider our method to be the first step towards building large-scale generative grasp and manipulation models from multi-view data.

Chapter 4

MANUS: Markerless Grasp Capture using Articulated 3D Gaussians

In the previous chapter, we discussed RealGrasper, a novel grasp generation method. Despite its promising results, RealGrasper had limitations due to the difficulty in calculating contacts because of its implicit representation. Additionally, both training and inference with this method were slow.

In this chapter, we explore an alternative method called Gaussian Splatting [19], which features an explicit representation using Gaussian and offers rapid optimization and inference. We leverage this representation to address another challenge in the literature: markerless grasp capture for accurate contact estimation.

To this end, we present MANUS, a method for Markerless Hand-Object Grasp Capture using Articulated 3D Gaussians. We build a novel articulated 3D Gaussians representation that extends 3D Gaussian splatting for high-fidelity representation of articulating hands. Since our representation uses Gaussian primitives, it enables us to efficiently and accurately estimate contacts between the hand and the object. For the most accurate results, our method requires tens of camera views that current datasets do not provide. We therefore build MANUS-Grasps, a new dataset that contains hand-object grasps viewed from 53 cameras across 30+ scenes, 3 subjects, and comprising over 7M frames. In addition to extensive qualitative results, we also show that our method outperforms others on a quantitative contact evaluation method that uses paint transfer from the object to the hand.

4.1 Introduction

Every day, the average person effortlessly grasps more than a hundred different objects [25, 26]. This seemingly routine act of grasping poses a significant challenge for machines, as is evident from the extensive research on this topic in computer vision [28] and robotics [27, 87]. High-fidelity capture of natural human grasps could unlock new applications in areas like robotics and mixed reality, but this challenging problem first requires us to accurately **estimate the contact** between the hand and the object [12].

Previous work has addressed this problem by using gloves or special sensors [41, 61], but these devices are cumbersome and restrict hand movement. Therefore, a large body of work has focused on **markerless grasp capture** using one or more cameras [4, 42, 43, 46, 53].

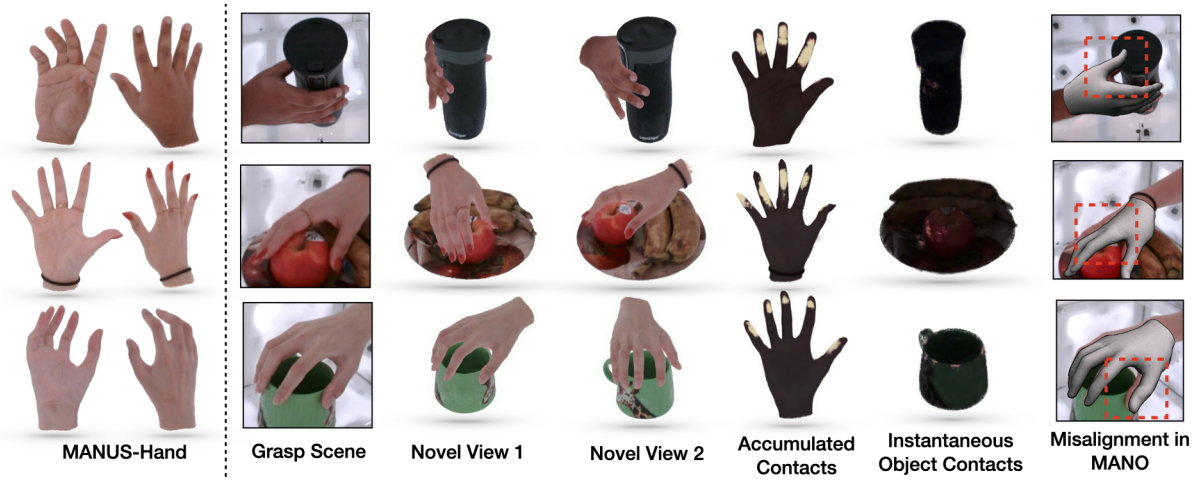

Figure 4.1: We introduce MANUS, a novel markerless approach for capturing grasps by employing an articulated 3D Gaussian representation to accurately model hand shapes. This approach improves contact estimation accuracy in comparison to other template-based approaches when evaluated against ground truth contacts.

Most of these methods use skeletons [46], meshes [43], or parametric models [2, 63] to model the hand and object. Although these representations are flexible and easy to use, they often cannot accurately model hand shape resulting in reduced contact accuracy (see 4.1). Recently, articulated neural implicit representations [17, 76, 88] have been proposed as alternatives, but modeling contact in implicit representations is challenging and requires expensive sampling.

To overcome these limitations, we introduce **MANUS**, a method for markerless grasp capture using articulated 3D Gaussians. The key component of MANUS is a 3D Gaussian splatting [19] approach to build **MANUS-Hand**, an articulated hand model composed of 3D Gaussians that make it faster to optimize and infer than many implicitly-represented models. Similarly, we also capture the object using static 3D Gaussians. Since both MANUS-Hand and the object are modeled using Gaussians primitives with explicit positions and orientations, we can efficiently compute both *instantaneous* and *accumulated* contacts between them. When trained on datasets with tens of camera views, our method can accurately capture grasps since 3D Gaussians promote accurate pixel-level alignment resulting in more precise shape and contact estimation compared to existing methods.

Previous datasets [3, 5, 12, 37, 46, 51, 89, 90] have been instrumental in addressing the grasp capture problem but (1) they use specialized hardware (heat-sensitive cameras [12], or markers [3]) to capture hand-object grasps, making it hard to scale, (2) RGB camera-only datasets [4, 5, 16, 53], contain only a few views with occlusions making it hard to learn accurate contacts, and (3) they rely on the parametric models or skeletons to estimate contacts resulting in inaccurate contacts. **Our main insight is that accurate contact modeling is much easier with a large number of camera views that reduce the effect of (self-)occlusions.** Therefore, we curated a one-of-a-kind real-world multi-view RGB dataset, **MANUS-Grasps**, comprising over **$\tilde{7}$M frames** captured using 50+ high-framerate cameras, providing

a full 360-degree coverage of grasp sequences occurring in over 30 diverse everyday scenarios. In addition, this dataset contains 15 evaluation sequences that employ wet paint on objects to leave a contact residue on the hand [91] providing a natural way to evaluate contact quality without additional equipment or annotation. We show extensive experiments ablating and justifying different components of MANUS-Hand, as well as the MANUS grasping method. In addition, we also provide a new metric of contact quality to assess the performance of MANUS against template-based methods. While our method is not designed for photorealism, we observe that the captured grasping sequences are comparable in visual quality to the best implicit hand models.

To summarize, our contributions include:

- **MANUS-Hand**, a new efficient representation for articulated hands that uses 3D Gaussian splatting for accurate shape and appearance representation.
- **MANUS**, a method that uses MANUS-Hand and a 3D Gaussian representation of the object to accurately model contacts.
- **MANUS-Grasps**, a large real-world multi-view RGB grasp dataset with over $\tilde{7}$M frames from 50+ cameras, providing full 360-degree coverage of grasps in over 30 diverse everyday life scenarios.
- A unique and novel approach to validate contact accuracy using **paint transfer** between the object and the hand.

4.2 Related Work

Representations: Skeletons and collections of shape primitives were some of the first representations to be used for hand–object interaction modeling [42,61], but these representations are often not accurate enough for contact estimation. Meshes [43] and parametric models [2,63] are currently the most popular alternatives but can also be misaligned with observations due to their lower-dimensional representation (see 4.1).

Coordinate-based implicit neural networks, or neural fields [1], have shown great promise in accurately modeling shape and appearance in static scenes [17, 19, 21, 39, 64–71] as well as dynamic scenes [92–97]. Several methods specifically address articulated shapes [18] like human bodies [18,72–75], or hands [76, 88, 98–100]. However, they use representations that are inefficient for sampling and contact estimation. In contrast, we propose a new articulated neural field representation that extends 3D Gaussian splatting [19] to hands enabling efficient training/inference and contact estimation.

Hand-Object Interaction Capture: Previous work has attempted to model hand-object interactions using skeletons [16, 46], or customized meshes [43] as the hand representation without explicitly estimating contacts. Most other work [3–5, 37, 89] uses MANO in combination with mocap, or one or more camera views. While it becomes easier to estimate contact with a parametric mesh model, misalignments are still common (see 4.1). To overcome the difficulty of accurate contact estimation, some

methods resort to physical simulation [8, 9, 101], but these are limited to synthetic grasps only. In contrast, we propose a template-free articulated 3D Gaussian splatting model that provides a natural way to estimate accurate contacts.

Grasp Datasets Datasets for human grasps are challenging to obtain because they need specialized hardware, extensive annotation, and significant post-processing to make them useful. Some datasets use markers or special gloves to track the hand and object [34, 36, 40, 41] but this hinders natural hand motion and introduces changes in image appearance. Synthetic datasets [37, 48, 49] suffer from a domain gap that makes it challenging to generalize to real data. Therefore, work has focused on manual annotations [42–45], optimization [46], or automatic annotation [4, 47] from RGB or depth. Many of these datasets provide only 3D hand poses and lack information about contacts. Other datasets like InterHand2.6M [50, 51] are limited to hands only without any objects, while others [52] focus on 2D understanding only. Addressing these limitations, HOnnotate [46] introduces a markerless system for automatically annotating frames across 77K frames. However, the variety of objects and grasps in this dataset is somewhat limited. ContactDB [12] and ContactPose [53] address this limitation targets a broader variety of grasps. While ContactDB is captured using thermal imaging, ContactPose uses multi-view RGB-D data. Nonetheless, both methods are restricted to 3D hand poses, use non-realistic objects, and lack sufficient views for neural fields.

In contrast, we introduce MANUS-Grasps that includes diverse grasps from 50+ cameras capturing at 120 FPS specifically to support neural field methods. In total, we provide over $\tilde{7}$M frames with ground truth camera poses, segmentation, and estimated contacts.

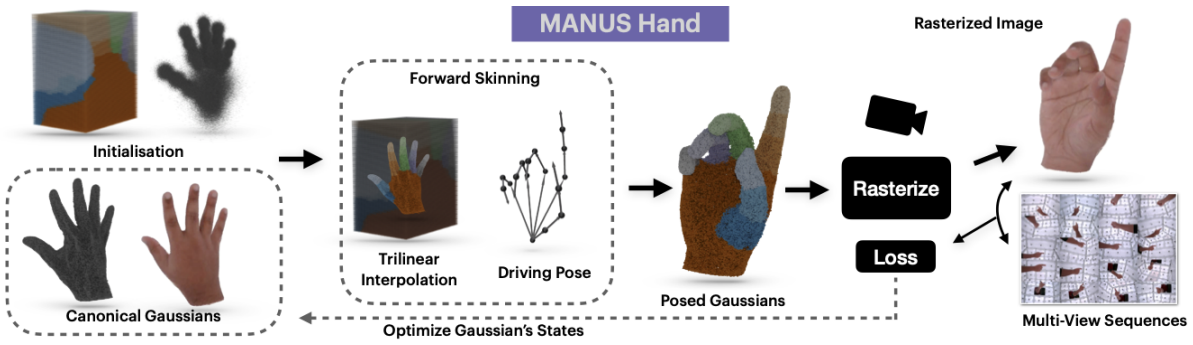

Figure 4.2: **MANUS-Hand** is a template-free, articulable hand model learned from multi-view hand sequences which utilizes 3D Gaussian splatting representation for accurate modelling of the shape and appearance of hands.

4.3 Background

We briefly summarize recent advances in modeling radiance fields of static and dynamic scenes using 3D Gaussians [19,97,102]. Our method (see 4.4) extends the 3D Gaussians representation to articulated objects like the hand, and for grasp capture.

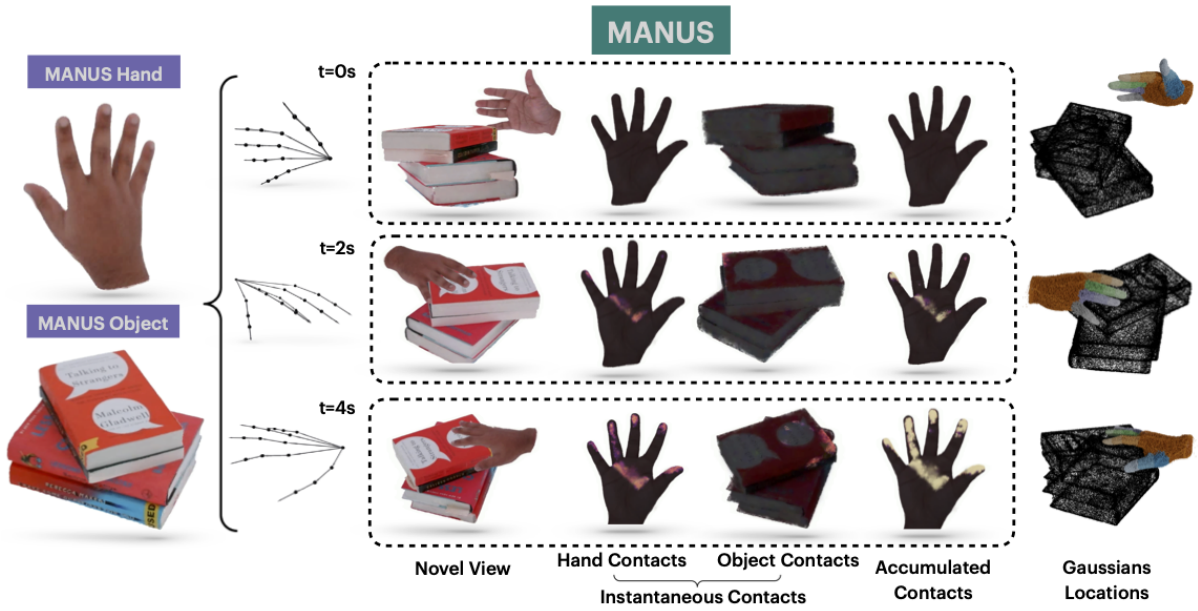

Figure 4.3: **MANUS** leverages a driving pose to get MANUS-Hand in grasp scene. It is combined with an object model to get instantaneous and accumulated contacts between the two.

Static 3D Gaussians Given multi-view images and a sparse point cloud of the scene, a set of 3D Gaussian primitives can be defined across world space $x \in \mathrm{R}^{3\times1}$ as,

$$G(x) = e^{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)},$$

here each Gaussian primitive has 3D position (μ), opacity, anisotropic covariance matrix (Σ), and spherical harmonic (SH) coefficients. During the training of the radiance field, the properties of the initial 3D Gaussians are optimized together with a tile rasterizer [19] with the objective of minimizing pixel loss.

Dynamic 3D Gaussians The 3D Gaussians approach has recently been extended to dynamic scenes [19, 102]. [102] introduces a deformation field that tracks the Gaussian position across timesteps. Similarly, [97] enable Gaussians to move and rotate over time while maintaining their color, opacity, and size. While these methods can capture dynamic and deformable scenes, they do not provide a way to control dynamic motion, e.g. , using a skeleton. Furthermore, in these methods, Gaussians are free to move within the scene without any restrictions, which isn't suitable for representing hands due to their kinematic structure. An articulated 3D Gaussians representation would be advantageous for grasp capture since it would enable low-dimensional skeleton-based control of the hand.

4.4 Method

MANUS aims to perform markerless capture of human hand grasps by accurately estimating the shape, appearance, and contacts between the hand and the object from multi-view RGB videos. We

Dataset	#N Images (Views)	Annot. Type
w/o Contacts Annotation		
H2O-3D [90]	76k (5)	multi-kinect
FHPA [41]	105k (1)	magnetic
HOI4D [89]	2.4M (1)	single-manual
FreiHand [51]	37k (8)	semi-auto
HO3D [46]	78k (1-5)	multi-kinect
DexYCB [4]	582k (8)	multi-manual
ARCTIC [5]	2.1M (9)	mocap
w/ Estimated Contacts Annotation		
ContactPose [53]	2.9M (3)	multi-kinect
GRAB [3]	- (-)	mocap
H2O [16]	571k (5)	multi-kinect
w/ Ground-Truth Contacts Annotation		
MANUS-Grasps (Ours)	7M (50+)	multi-auto

Table 4.1: Dataset Comparison of existing Real World Datasets. The hands in previous datasets are represented by skeleton and MANO. Different from other works, we use Gaussian to model the hand. The keyword "single/multi-manual" denotes whether single or multiple views being used to annotate manually.

achieve this by combining MANUS-Hand with an object model, both represented as 3D Gaussians, enabling us to compute contacts more efficiently than sampling-based implicit representations. 4.3 provides an overview of our method.

4.4.1 MANUS-Hand

Our template-free, articulated hand model MANUS-Hand adopts 3D Gaussian splatting as the representation for accurate shape and appearance modeling of hands. Our model can be trained on sequences from any multi-view dataset to build an articulable hand model at any novel pose.

Representation MANUS-Hand (see 4.2) is composed of a skeleton with 21 bones and has 26 degrees of freedom (check supplementary for bone-specific DOFs). We built a custom pose estimation pipeline that uses AlphaPose [20] to estimate the 3D joint positions followed by an inverse kinematics fit (check supplementary). Since bone lengths can vary among different individuals, we estimate these lengths from the dataset and adjust the skeleton accordingly. The unique shape and appearance of a person's hand in a canonical pose are determined by the states of 3D Gaussians, i.e. , positions μ, covariances

Σ, opacities α, and spherical harmonics coefficients ϕ. The covariance of each Gaussian in the canonical space is further defined as $\Sigma = RSS^TR$, where R and S denote the rotation and scaling of the Gaussians.

Optimization A unique MANUS-Hand is optimized separately for each subject from a dense multi-view dataset containing approx 20 hand poses. To initialize Gaussian states in MANUS-Hand, we set their means to be points on a normal distribution centered at the midpoint of each bone in a *canonical* hand pose, with the distribution's standard deviation adjusted to match the bone's length (as shown in 4.2). We follow a similar protocol as [19] to initialize the covariances, opacity, and SH coefficients.

To get the Gaussian positions in the posed space, forward kinematics and linear blend skinning is applied to the canonical Gaussians. One way to obtain skinning weights is to assign MANO weights [2] directly to the closest Gaussians. However, this approach results in artifacts because Gaussians could move in unpredictable ways during training leading to mismatched skinning weights (visualized in ablation study) To address this, we create a canonical grid inspired by Fast-SNARF [103]. Skinning weights are then allocated to grid voxels using the nearest neighbor method, termed as grid weights. Now to obtain the skinning weights for the queried Gaussians W in the canonical space, trilinear interpolation of these grid weights is performed. We calculate the transformed Gaussian positions using a per-bone transformation matrix, denoted as T_b and linear blend skinning: $T_g = WT_b$, $\mu_p = T_g\mu$, where μ_p represents the location of Gaussians in the posed space, and T_g represents the transformation matrix for each Gaussian. To compute the covariance of the Gaussians in the posed space, it is transformed using a rotation matrix R_g, derived from T_g. This is expressed as $\Sigma_p = R_g\Sigma R_g^T$. Regarding the appearance, we optimize spherical harmonics coefficients for each Gaussian ϕ_g in the canonical space. To get the colors in the transformed or posed space, the view direction from posed space ν_p^g is first converted to the canonical space ν_c^g as $\nu_c^g = T_g^{-1}\nu_p^g$, using T_g for each Gaussian. After this step, we use these transformed view directions μ_c^g to query the spherical harmonics coefficients in canonical space and get corresponding RGB colors for each posed Gaussian. To get the final image rendering, all Gaussian states currently in the posed space are used as inputs to a differentiable rasterizer [19], denoted as $\mathcal{R}$

$$\mathcal{I} = \mathcal{R}(\mu_p, \nu_c, \Sigma_p, \alpha, \phi), \tag{4.1}$$

where $\mathcal{I}$ is the rendered image. During optimization, the Gaussian states are optimized using to minimize pixel loss on the posed hand. To optimize all Gaussian states, we impose a rendering loss $\mathcal{L}_1 = \|\hat{\mathcal{I}} - \mathcal{I}\|$ and structural similarity [104] loss $\mathcal{L}_{SSIM}$ between synthesized image $\mathcal{I}$ and ground truth image $\hat{\mathcal{I}}$ of the posed hand. To further improve the perceptual quality of the synthesized images, we add an additional perceptual loss $\mathcal{L}_{perc}$ [105].

To avoid highly anisotropic Gaussians that could cause artifacts in the contact rendering, we incorporate an isotropic regularizer which ensures optimized Gaussians remain as isotropic as possible. If $\min_s \in R^3$ and $\max_s \in R^3$ are the minimum and maximum scale of the optimized Gaussians, then

isotropic regularizer $\mathcal{L}_{iso}$ is defined as

$$\mathcal{L}_{iso} = (\frac{\min_s}{\max_s} - s)^2, \tag{4.2}$$

where s is set to be 0.4. Our final loss function is $L_h = \alpha\mathcal{L}_1 + \beta\mathcal{L}_{SSIM} + \gamma\mathcal{L}_{perc} + \delta\mathcal{L}_{iso}$.

Inference Once the Gaussian states are optimized, we can drive MANUS-Hand using a skeleton obtained from our pose estimation pipeline (check supplementary). Given a novel pose during the inference, MANUS-Hand outputs the transformed Gaussians as well as the rendered image from a particular view.

4.4.2 MANUS: Grasp Capture

While MANUS-Hand enables high-fidelity articulated hand modeling, it is not designed for capturing grasps and contacts. To capture grasps, we need a representation of the object as well as a method to estimate contacts.

Object Representation For accurate representation of objects, we build a non-articulated Gaussian representation following 4.4.1 with some improvements to maintain geometric consistency and accuracy. To prevent floaters during optimization, we prune outlier Gaussians by projecting on image and culling if they lie outside the object mask.

Grasp Capture To capture the grasp in a particular sequence, we first articulate MANUS-Hand using the estimated hand pose. We then construct the object model as described above. Next, we combine both hand and object Gaussians. More specifically, if G_h and G_o are the hand Gaussians and object Gaussians in the grasp scene, we simply concatenate the Gaussians $G_f = \{G_o, G_h\}$. Because we use Gaussian Splatting, it allows such a concatenation operation naturally – this would not be possible with implicit representations [18, 76, 88]. As the rasterization module only requires a set of Gaussians and their states, we can seamlessly merge hand and object Gaussians for every frame. The final grasp image is given by a rasterized composition of these Gaussians using 4.1.

Contact Estimation The contact map is calculated based on the proximity in 3D space between hand and object Gaussian positions. For each Gaussian on the hand, we find the closest Gaussian on the object. This pair is considered to be in contact if their distance is less than a certain threshold, and the same applies when assessing contact from the object's perspective. Specifically, if G_h represents the Gaussians on the hand and G_o those on the object in the posed space, then the 3D contact map between them is defined as:

$$C = \begin{cases} d(G_h, G_o), & \text{if } d(G_h, G_o) < \tau \\ 0, & \text{otherwise} \end{cases},$$

where d represents the pairwise Euclidean distance between the Gaussian locations. A contact is considered to have occurred if this distance is less than τ, which is the predefined threshold for contact. We then use this method to estimate two kinds of contact maps on the hand and object: (1) an **instantaneous contact map** that denotes contact at a specific timestep, and (2) an **accumulated contact map** that de-

notes contact after the grasping has concluded. To get the accumulated contact map C_{acc} we simply add the previous frame's accumulated contact map to current frame. For rendering contact maps, we employ 4.1 using the contact distance as the color value of each Gaussian.

4.4.3 MANUS-Grasps

For our grasp capture method to work well, a key requirement is a multi-view RGB dataset with tens of camera views that help resolve self-occlusions. Many prior datasets and 4.1) contain multi-view images or video of hand grasps [34, 46, 47], but none have the large number of views needed to support neural field representations or are limited to hands only [50]. We therefore present MANUS-Grasps, a large real-world multi-view RGB grasp dataset with over $\tilde{7}$M frames from 50+ cameras, providing full 360-degree coverage of grasp sequences comprising of 30+ diverse object scenes.

Capture System Our customized data capture setup consists of 53 RGB cameras uniformly located inside a cubical capture volume with each cube face consisting of 9 cameras. The sides of the cube are illuminated evenly using LED lights. Each RGB camera records at 120 FPS with a resolution of 1280×720. The cameras are software synchronized with a frame misalignment error of no more than 3 ms. The multi-view system is calibrated for camera intrinsics and extrinsics using COLMAP [77, 78] with fiducial markers on the walls.

Capture Protocol Our capture protocol consists of four steps. First, we recorded multi-view videos of a subject's right hand as they performed a brief articulating movement. Next, we capture only the object without the hand. Then, without moving the object, we record multi-view videos of the subject's hand grasping the object. We repeat this process 30+ times per subject with 2-5 grasps per object scene. For evaluation sequences, we additionally capture a canonical pose at the end to record accumulated contacts seen in the transferred paint (see below).

Ground Truth Contact A unique feature of our dataset is the capture of 15 evaluation sequences where the object has wet paint during the grasp [91]. As a result, paint is transferred to the hand resulting in visual evidence of contact. This contact mark is a physically accurate representation of the true (accumulated) contact between the hand and the object making it the true ground truth (even methods like [12] suffer from heat dissipation). We chose a bright green paint to enable automatic segmentation thereby creating a **gold standard** for contact evaluation.

Data Annotation MANUS-Grasps also provides 2D and 3D hand joint locations along with hand and object segmentation masks. We obtain the joint locations from AlphaPose [20] followed by 3D triangulation and inverse kinematics [22]. We impose constraints to limit the degrees of freedom and joint angles for the rotation of the bones. To achieve temporal smoothness for the sequence, we apply the 1€ Filter [24] on the estimated parameters. To segment the hand and object from the background, we use the Segment Anything Model (SAM) [106] followed by fitting an Instant-NGP model [21] to extract a binary mask to ensure multi-view consistency.

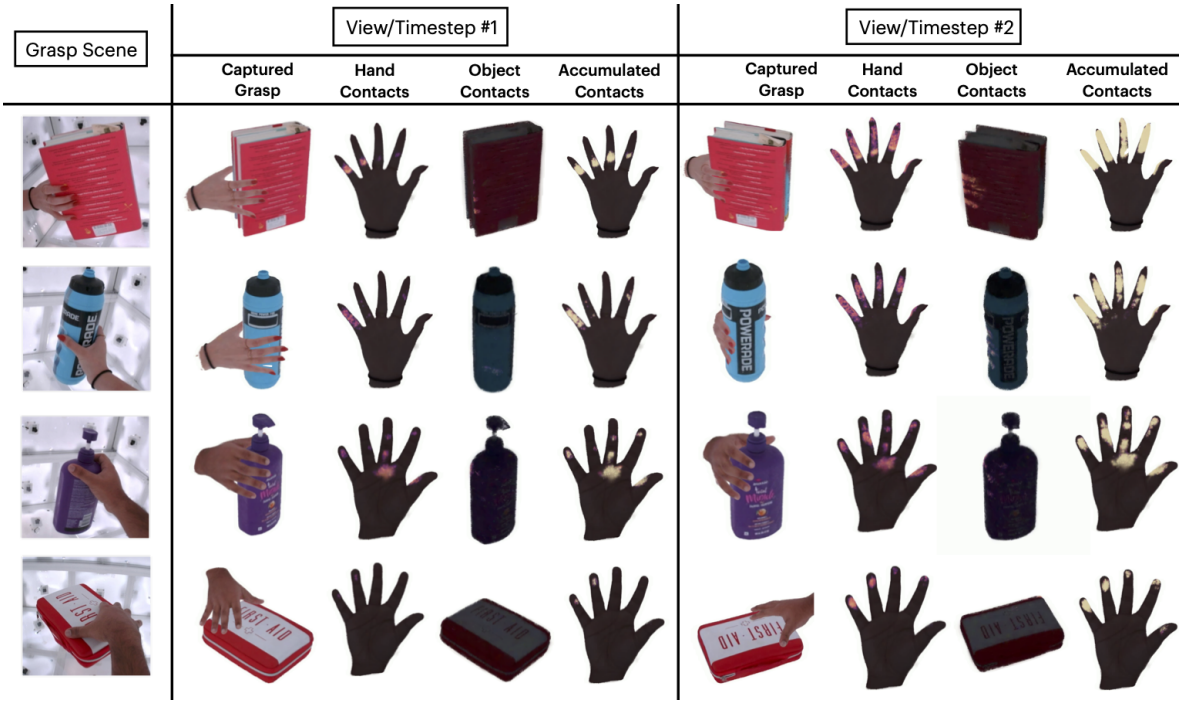

Figure 4.4: Here we show our contact estimation results on novel views for a variety of objects. We show both instantaneous and accumulated contacts for the hand in a canonical pose. Best viewed zoomed.

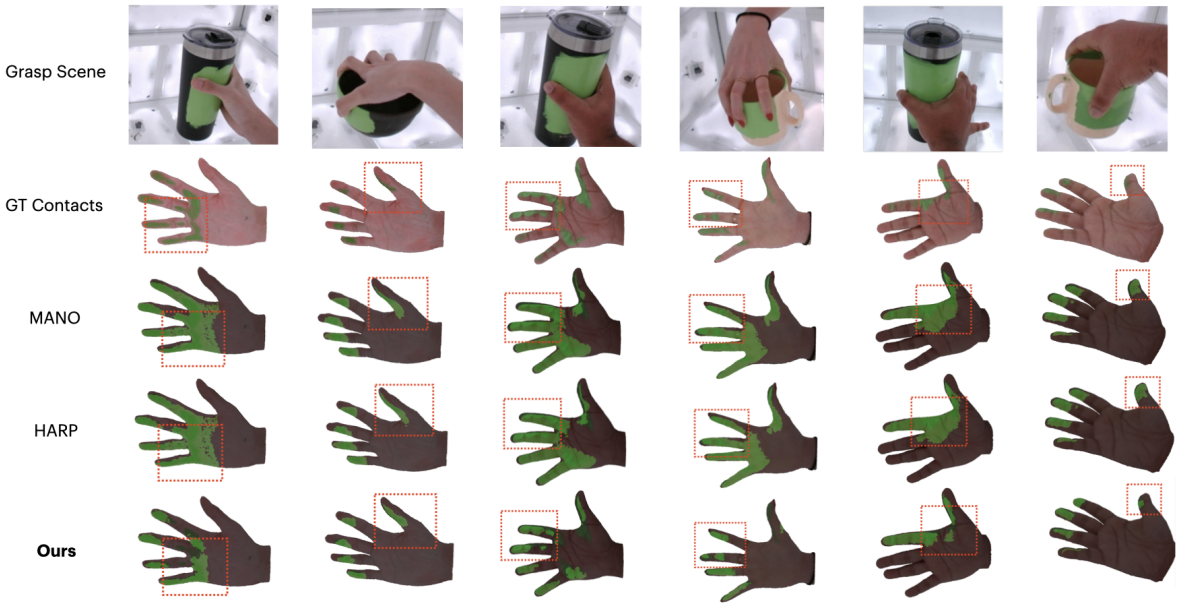

Figure 4.5: **Contact Comparisons**: We compare accumulated contacts of MANUS with that of MANO and HARP on ground truth contacts from MANUS Grasps dataset. It's visible that our contacts are far more accurate and closer to the actual ground truths.

4.5 Experiments and Results

In this section, we show qualitative and quantitative results from our method. Our goal is to evaluate both the MANUS-Hand and the MANUS grasp capture method, and compare with existing methods.

4.5.1 Evaluating MANUS-Hand

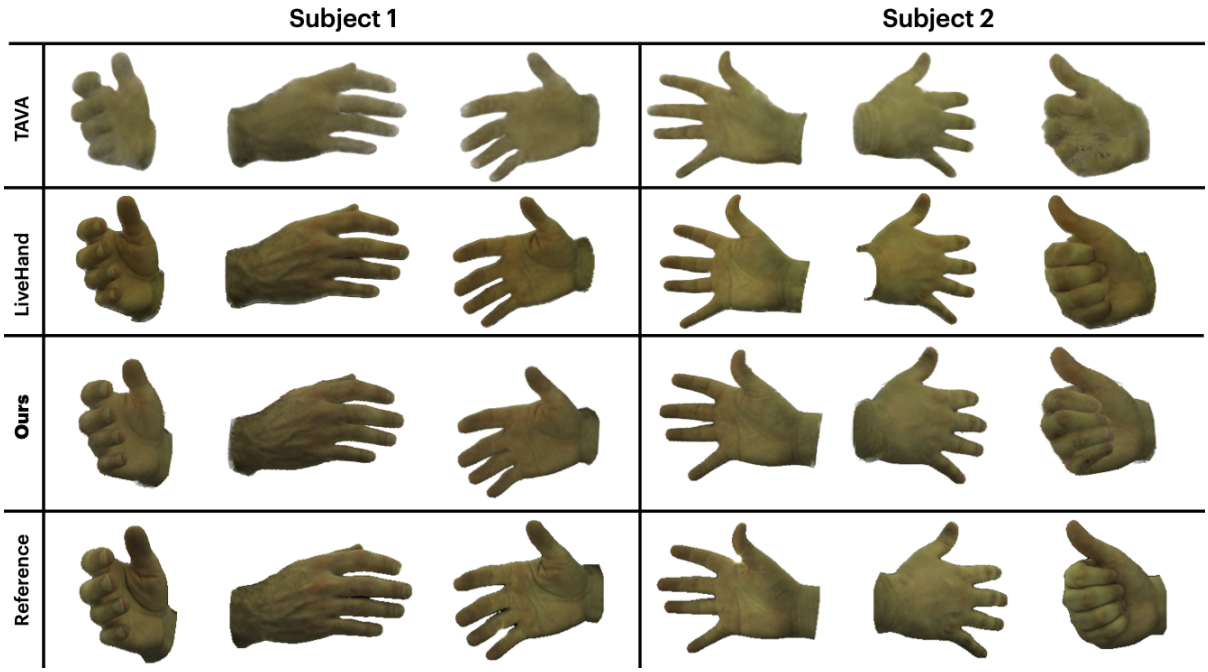

Figure 4.6: Qualitative comparison of MANUS-Hand with LiveHand [88] and TAVA [18]. It's noteworthy that our renderings closely resemble those of LiveHand and surpass TAVA in quality, even in the absence of any components designed to enhance photorealism.

We first show results and experiments related to MANUS-Hand only. We quantitatively as well qualitatively assess the visual quality of our hand model with the current state-of-the-art method LiveHand [88] and TAVA [18].

Metrics, Dataset & Setup: We assess the visual quality of our hand model using PSNR, SSIM, and LPIPS metrics (where higher scores indicate better performance) on the Interhand2.6M dataset, as shown in Table 4.6. We used two subjects from Interhand2.6M (Capture0 and Capture1), focusing on the "ROM07-RT-Finger-Occlusions" sequence from the test set. We allocate 75% of the data for optimizing and use the remainder for evaluation.

Quantitative Evaluation: MANUS-Hand is not specifically designed for photorealism since we leave out ambient occlusion and shadow mapping and focus only on geometric accuracy. As shown in Table 4.6, our results outperforms TAVA however LiveHand emerges as the best in terms of the evaluated metrics (PSNR/LPIPS), which significantly penalize the absence of ambient occlusion and shadows

(also mentioned by [18]). *We want to emphasize that our primary goal is not to surpass existing hand models in terms of visual quality. Instead, our focus is on accurate contact estimation.* LiveHand and TAVA both learn implicit volumetric density field which makes calculating contact maps complicated & expensive, whereas our Gaussians-based approach is more efficient. The comparison with LiveHand and TAVA is intended to demonstrate our comparable visual quality despite not being designed for it.

Qualitative Evaluation: We conducted a qualitative comparison of our MANUS-Hand with TAVA [18] and LiveHand [88], as shown in Figure 4.6. The quality of our renderings is superior to TAVA [18] and is on par with that of LiveHand. In conclusion, despite not being tailored for photorealism, our method demonstrates substantial potential for application in photorealistic contexts.

4.5.2 Evaluating Grasp Capture

Next, we evaluate our MANUS method for grasp capture. In this chapter, we assume that direct contact between the hand and the object is the primary mode of grasping (we ignore indirect grasping through tools). Therefore, the goal of grasp evaluation is to objectively measure the accuracy of contacts. We compare three methods: (1) MANO [2] fitting methods, (2) HARP [100], and (3) our MANUS model.

Metric, Dataset & Setup: In our experiments, we use the wet-paint transfer method [91] to accurately collect ground truth accumulated contacts (see Section 4.4.3). After grasp completion, users are instructed to return to a canonical post-grasp pose. In this pose, the green paint residue in the grasping hand is automatically segmented and 2D contact maps are rendered from 10 different views (details in supplementary) using [21]. We then assess the quality of grasps estimated by different methods using the Intersection over Union (IoU) and F1-score metrics. All experiments use 15 sequences of our wet-paint evaluation sequences. We set the distance threshold $\tau = 0.004$ for contact estimation for all methods. For a fair comparison, we subdivide the meshes of MANO and HARP from 778 to 49,000 vertices before estimating contact. For estimating contact masks in all methods, we utilize the 'gray' color map [107] on the distance map. The contact masks for MANUS are rendered using [19], while for the other two frameworks, they are rendered using the emission shader in Blender. It's noteworthy that MANUS **consistently outperforms** the others in the contact metric across all three subjects as shown in Table 4.2.

Qualitative Evaluation: We also present a qualitative comparison of our contact results against those obtained using MANO and HARP in Figure 4.5. Our method shows a more accurate representation of the contact area, closely matching the actual contact masks, unlike the over-segmentation observed in MANO and HARP methods. Although our method outperforms others, we note that there is still significant room for improvement on our dataset for future methods to address.

Method	Subject1	Subject2	Subject3
mIoU ↑			
MANO	0.161	0.135	0.208
HARP	0.173	0.148	0.224
Ours	**0.206**	**0.152**	**0.275**
F1 score ↑			
MANO	0.270	0.228	0.338
HARP	0.28875	0.2474	0.361
Ours	**0.335**	**0.251**	**0.424**

Table 4.2: Comparison of MANUS grasp capture approach with MANO and HARP on contact metric. Note that, we perform consistently better in both metrics.

4.6 Ablation Study

4.6.1 MANUS-Hand

Initialization of Skinning Weights: We observe that the choice of method used to initialize skinning weights significantly influences the performance of our hand model. As demonstrated in Figure 4.10 (a), initializing skinning weights directly onto Gaussians using a nearest neighbor approach, as opposed to grid initialization, leads Gaussians to move erratically and shift towards an unrelated bone. Consequently, this misalignment results in artifacts, where skinning weights are incorrectly allocated to the wrong bone, causing the position to be associated with the incorrect bone. The impact of this method of initialization is presented both quantitatively and qualitatively in Table 4.3 and Figure 4.9.

Ablation on LPIPS loss: We observed that LPIPS loss improves the quality of renderings and maintain consistency across views. We also demonstrate that LPIPS loss function improves the overall visual quality of our hand model qualitatively at Figure 4.9 and quantitatively at Table 4.3.

Figure 4.7: We display a comparison of the pixel misalignment between projected Gaussians and the MANO mesh against a reference image.

Alignment with image pixels: We now demonstrate the pixel-alignment results of MANUS-hand and MANO in Figure 4.7. Due to inherent design and photo-metric losses, our hand representation is pixel-aligned to reference image, resulting in reduced alignment as compared to that of MANO.

Benchmarking MANUS Grasp scenes: We also evaluate our MANUS Hand and Object method in Table 4.4 using the data included in the MANUS Grasp dataset. The well-lit scenes and the absence of harsh shadows in our dataset lead to improved evaluation metrics when compared with those of the InterHand2.6M dataset.

4.6.2 MANUS Grasp Capture

Affect of the number of Gaussians in contact map rendering: We show in Figure 4.10(b) that the quality of accumulated 2D contact maps deteriorates when the number of Gaussians is reduced. Therefore, in our experiments, we make sure to densely initialize Gaussians for both objects and hands.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Test time (s) ↓
w/o grid	26.108	0.987	0.0729	**0.0082**
w/o lpips	25.92	0.986	0.074	0.043
Ours	**26.328**	**0.9872**	**0.0688**	0.043

Table 4.3: Ablation on weight initialization approach and choice of LPIPS loss. Our design approach improve all visual quality metrics.

4.7 Implementation Details

Our method was implemented in Python using the PyTorch Lightning [108] framework. All experiments were conducted using a single Nvidia RTX3090 GPU with gradient accumulation for 4 iterations. The weights of the different loss function terms - α, β, γ and δ - were experimentally determined and set at values of 0.7, 0.1, 0.1, and 0.1, respectively. In all our experiments, we chose a grid size of 256x160x142 around the canonical hand skeleton for storing the skinning weights initialized from MANO [2]. MANUS-Hand is initialized with 30K Gaussians per bone, amounting to 900K Gaussians in total. After training, this number is pruned and filtered down to approximately 300K.

4.8 MANUS-Grasps Dataset Details

Bone length estimation: We first use the [20] to acquire 2D keypoints for every frame and view. These keypoints are then triangulated into 3D keypoints using the [109]. With these triangulated keypoints, we determine the bone lengths for each subject. Specifically, we average the 3D keypoints across all grasp sequences and then adjust the length of the skeleton accordingly.

Categories	PSNR ↑	SSIM ↑	LPIPS ↓
Mugs	43.08	0.999	0.002
Bottles	38.17	0.997	0.008
Fruits	39.57	0.998	0.005
Utensils	38.25	0.994	0.009
Misc	38.79	0.995	0.008
Colored	42.38	0.999	0.004
Bags	38.44	0.994	0.011
Jars	40.66	0.999	0.005
Books	36.17	0.998	0.015
Tech	38.81	0.995	0.007
Hand1	28.34	0.995	0.031
Hand2	29.94	0.998	0.029
Hand3	29.71	0.997	0.027

Table 4.4: Here we benchmark MANUS-hand and object method on MANUS Grasp scenes.

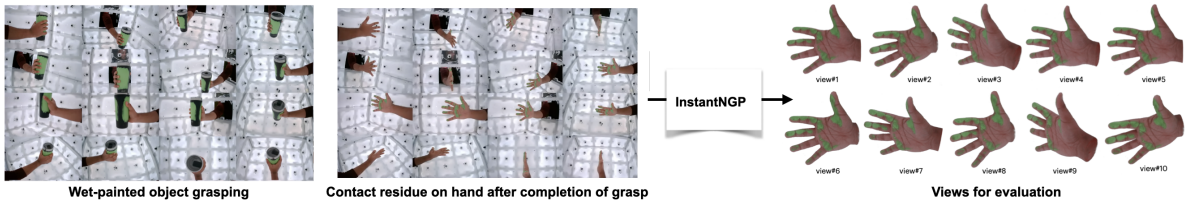

Figure 4.8: Here, we show the approach we used to obtain the ground truth contacts for the evaluation sequences. On the far right, we display all 10 views of one evaluation sequence for the quantitative assessment of grasp capture.

Camera Views	Subject1	Subject2	Subject3
mIoU ↑			
5	0.147	0.140	0.214
10	0.164	0.145	0.256
20	0.176	0.142	0.261
Ours (30+)	**0.206**	**0.152**	**0.275**
F1 score ↑			
5	0.244	0.235	0.343
10	0.266	0.242	0.401
20	0.271	0.240	0.410
Ours (30+)	**0.335**	**0.251**	**0.424**

Table 4.5: Here we show empirical findings demonstrating the decline in contact metric as the number of camera views decreases, leading to increased susceptibility to self-occlusions.

Segmentation: For every segmentation task, we employ a combined approach utilizing InstantNGP [21] and SAM [106]. Initially, the scene is segmented using the text-based SAM technique. Following

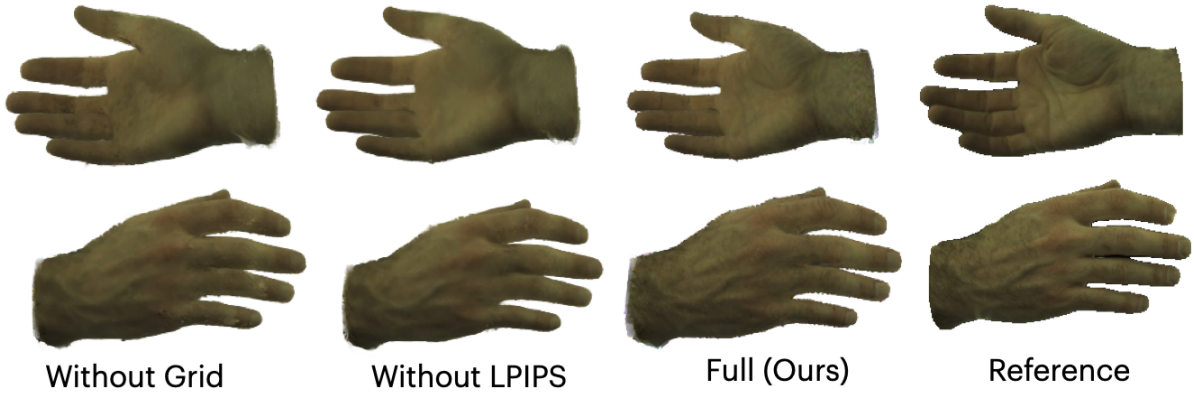

Figure 4.9: **Hand Ablation**: We perform ablation on the grid initialization of the skinning weights and the choice of LPIPS loss function. Clearly our approach is better in terms of visual appearance.

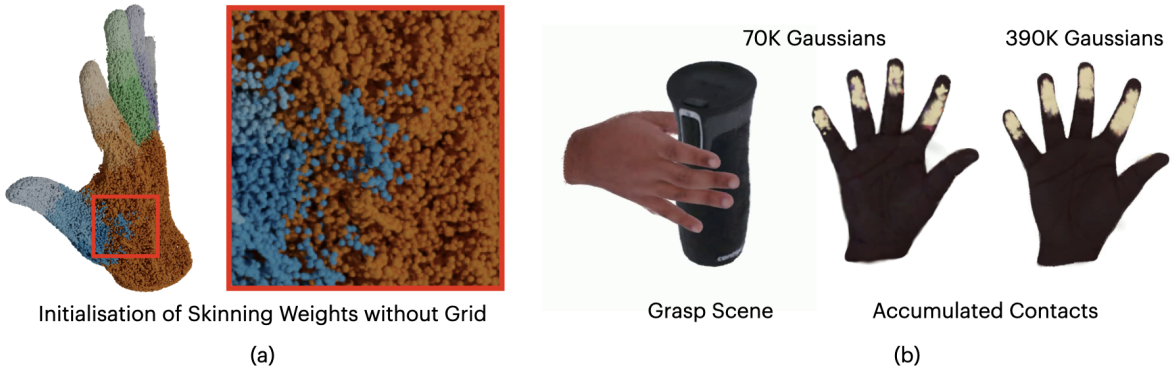

Figure 4.10: Here in **(a)** we show how initializing MANO weights without voxel grid allows the unstructured Gaussians to move erratically. In **(b)**, we show the affect on accumulated 2D contact renderings with change in the number of Gaussians.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Test time (s) ↓
TAVA	22.85	0.983	0.099	11.00
LiveHand	**31.16**	0.9818	**0.0278**	**0.022**
Ours	26.32	**0.9872**	0.068	0.049

Table 4.6: Here, we show comparison of MANUS-Hand on InterHand2.6M [50] dataset with LiveHand [88] and [18]. Note that our primary goal is to obtain accurate contacts, not visual quality.

this, we obtain a segmentation mask that maintains consistency across multiple views using InstantNGP. If the segmentation masks are found to be inadequate due to inaccurate predictions from the text-based SAM, the process is repeated until satisfactory results are achieved.

Ground Truth Contacts: In Figure 4.8, we illustrate the methodology used to gather ground truth contact data for our evaluation sequences. Initially, the object is coated with a layer of bright, wet paint. Following this, the object is grasped, resulting in the transfer of paint residue to the hand. After the

grasp is finalized, we document the pattern of contact residue left on the hand. To obtain the required viewpoints, we train [21] in the multi-view images and then select 10 distinct views for evaluation. We repeat this process for 15 different evaluation sequences for each subject.

Grip Aperture: The grip aperture [110] refers to the distance between the thumb and fingers when grasping or holding an object. It's an important concept in fields like ergonomics, rehabilitation, and robotics. Here in Figure 4.11, we plot the change of grip aperture with change in timestep for our dataset.

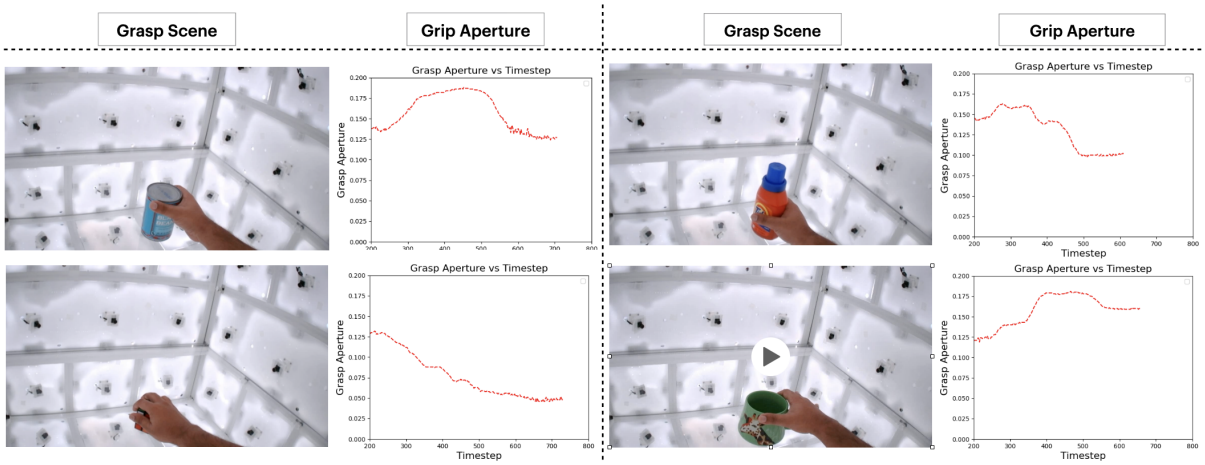

Figure 4.11: Variation of grip aperture with change in timestep while grasping.

4.9 MANO and HARP evaluation

Pose and Shape Estimation: We begin by estimating the shape and scale parameters of the MANO model for each subject. First, we obtain the mesh for every time-step by training [21] on multi-view images. Next, we refine the mesh through the use of MeshLab and Blender software to achieve a cleaned version. We employ an optimization framework akin to that used in [111], focusing on optimizing all MANO parameters, including angle, translation, shape, and scale for the first timestep. This optimization incorporates both keypoint loss (2.16) and point-to-surface loss [112] with the clean mesh. For subsequent sequences , we keep the shape and scale parameters unchanged, focusing solely on optimizing angles and translations through keypoint loss. To enhance the speed of convergence, we use the optimized parameters from the previous step as the starting point for new parameters.

To get better geometry than MANO we extend HARP [100] from monocular video setup to multi-view video setup. We start with already optimized MANO model (as mentioned above) and then optimize for the local displacement of the hand shape. We leverage the differentiable rasterizer, to optimize the HARP model based on the losses mentioned in [100].

Evaluation Setup: Please note that, we can't directly render contact maps for MANO and HARP in the same way as MANUS, which employs a Gaussian-based differentiable rasterizer. To obtain contact

maps for MANO and HARP, we initially allocate contact values to each vertex, followed by utilizing Blender's emission renderer to render the contact mask. For fair comparison, we increase the resolution of MANO and HARP vertices from 778 to 49,000.

Discussion: We also demonstrate the importance of dense camera views for accurate contact representation in Table 4.5 which shows the diminishing of contact metric as the number of camera view decreases. This finding is significant as it confirms our initial hypothesis that dense camera views are essential for accurate contact representation, helping to prevent self-occlusion scenarios.

Results: Finally, we show qualitative results in Figure 4.4, showcasing two different stages: one during the grasp process and another at the conclusion of the grasp.

For a comprehensive 360-degree view of the grasp capture, an in-depth ablation study, and details on the implementation, please refer to our supplementary materials.

4.10 Conclusion

In this work, we proposed MANUS, which introduced a novel articulated 3D Gaussians representation, which successfully bridge the gap between the accurate modeling of contacts in hand-object interactions and the limitations of current data capturing techniques. We introduced MANUS-Grasps, an extensive multi-view dataset captured from 50+ cameras, which offers an unprecedented level of detail and accuracy, covering a wide range of scenes, subjects, and frames. Overall, MANUS demonstrates remarkable potential in advancing the fields of robotics, mixed reality, and activity recognition, enabling the creation of more accurate robotic systems and enhanced virtual interactions.

Limitations and Future Work: While our focus in this chapter was on accurate contact estimation, we recognize that the complexity of hand dynamics in everyday life extends far beyond what we have explored. Our current focus has been on modeling single-hand grasping with static objects, without delving into the pose-dependent non-linear deformation caused by skin stretching. Additionally, hand-object manipulation for longer time-frames is unaddressed in this work and can be a interesting direction for future works. We also observe that there is room for improvement in the metrics we propose for future work. We also acknowledge the complexity and limited accessibility of our capture setup which motivates us to make dataset publicly available.

Chapter 5

Conclusion

In this chapter, we provide a brief conclusion on our contributions and the impact of proposed methods. We also discuss the potential future direction that can be explored.

5.1 Discussion

First, we introduced RealGrasper, a generative model that uses a conditional variational autoencoder. This model takes an initial 3D hand pose and object shape as input to predict the final grasp pose. Instead of relying on mesh-based representations, which may not accurately capture appearance and contact properties, this approach uses neural fields to model shape, appearance, and contact regions from various viewpoints. To support this, we created RealGrasp, a comprehensive dataset with 53-view RGB data, including over 362,000 frames, 11 different objects, 4 subjects, 17 free-hand sequences, and 20 grasps per object per subject. Although this method shows promise in generating grasps without 3D annotated data, it is computationally intensive during training, and calculating contacts within the implicit representation is challenging.

To address these issues, we explored an alternative method called Gaussian Splatting, which uses an explicit representation with Gaussians for faster optimization and inference. Using this representation, we developed MANUS, a method for Markerless Grasp Capture using Articulated 3D Gaussians. We also introduced MANUS-Hand which uses this novel representation for high-fidelity modeling of articulable hands. Additionally, we created MANUS-Grasps, a unique dataset featuring hand-object grasps captured from 53 cameras across 30+ scenes, 3 subjects, and over 7 million frames.

5.2 Impact

This thesis makes significant contributions to the field of hand-object interaction through the introduction of novel methods which doesn't require 3D annotated data. Through RealGrasper, we introduced a generative model that utilizes a conditional variational autoencoder to predict final grasp poses. This model incorporates initial 3D hand poses and object shapes, moving beyond traditional mesh-like representations to neural fields, and doesn't require any 3D annotated data. The introduction of

RealGrasp and MANUS-Grasps datasets significantly enriches the resources available for the research community. RealGrasp, with its 53-view RGB dataset and extensive frames and objects, provides a robust basis for training and evaluating hand-object interaction models. Similarly, MANUS-Grasps, with its 7 million frames across diverse scenes and subjects, offers an unparalleled resource for high-fidelity hand-object grasp modeling. The exploration and application of Gaussian Splatting for rapid optimization and inference marks a significant improvement in representation techniques. The MANUS method leverages articulated 3D Gaussians to provide a high-fidelity representation of articulating hands, offering a more efficient and scalable solution compared to previous methods. The methodologies and datasets developed in this thesis have broad applicability across various domains, including robotics, virtual reality, augmented reality, and animation. These contributions facilitate advancements in how machines understand and interact with their physical environment, potentially improving user experience and interaction in technology-driven fields.

5.3 Future Directions

Building on the contributions and findings of this thesis, several future directions can be explored to further advance the field of hand-object interaction.

Optimization of Computational Efficiency: One significant challenge with the RealGrasper model is its computational intensity during training. Future research could focus on optimizing the training process to make it more efficient. One simple solution is to integrate Gaussian splatting with the RealGrasper model for fast optimization and inference.

Contact metric: In MANUS, we introduced a novel way to collect hand-object interaction and also introduced a contact metric to evaluate the quality of contacts. Future efforts can be made to make this method more scalable in terms of capturing the ground truth contacts.

Extension to Dynamic Interactions: While this thesis primarily focuses on static grasp poses, future research could extend these methods to dynamic hand-object interactions. This involves modeling the temporal aspects of interactions, such as manipulation tasks, and developing models that can predict and adapt to changes in real-time.

Exploration of Multimodal Data: Incorporating additional data modalities, such as tactile feedback or force measurements, could provide a more comprehensive understanding of hand-object interactions. Future research could explore the integration of multimodal data to improve the accuracy and realism of the models.

List of Related Publications

Thesis Publications

- **Chandradeep Pokhariya*** , Ishaan Shah*, Angela Xing, Kefan Chen, Avinash Sharma, Srinath Sridhar; *RealGrasper: Learning Human Hand Grasping from Multi-View Images*; **Technical Report 2023**.
- **Chandradeep Pokhariya**, Ishaan Shah* , Angela Xing*, Zekun Li, Kefan Chen, Avinash Sharma, Srinath Sridhar; *MANUS: Markerless Grasp Capture using Articulated 3D Gaussians*; **Conference on Computer Vision and Pattern Recognition, 2024**.

Other Publications

- Cheng-You Lu*, Peisen Zhou*, Angela Xing*, **Chandradeep Pokhariya**, Arnab Dey, Ishaan Shah, Rugved Mavidipalli, Dylan Hu, Andrew Comport, Kefan Chen, Srinath Sridhar; *DiVa360-The Dynamic Visual Dataset for Immersive Neural Fields*; **Conference on Computer Vision and Pattern Recognition, 2024**.
- **Chandradeep Pokhariya***, Shanthika Naik*, Astitva Srivastava, Avinash Sharma; *Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization*; **SIGGRAPH Asia 2022 Technical Communications**. (Not claimed in this thesis.)
- Sai Sagar Jinka, Astitva Srivastava, **Chandradeep Pokhariya**, Avinash Sharma, P.J. Narayanan; *SHARP-Shape-Aware Reconstruction of People in Loose Clothing*; **International Journal of Computer Vision (IJCV'22)**.
- Astitva Srivastava, **Chandradeep Pokhariya**, Sai Sagar Jinka, Avinash Sharma; *xCloth-Extracting Template-free Textured 3D Clothes from a Monocular Image*; **ACM International Conference on Multimedia (ACMMM'22)**.

Bibliography

[1] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," *Computer Graphics Forum*, 2022.

[2] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, Nov. 2017.

[3] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "GRAB: A dataset of whole-body human grasping of objects," in *European Conference on Computer Vision (ECCV)*, 2020. [Online]. Available: https://grab.is.tue.mpg.de

[4] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield *et al.*, "Dexycb: A benchmark for capturing hand grasping of objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9044–9053.

[5] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, "ARCTIC: A dataset for dexterous bimanual hand-object manipulation," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[6] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *ArXiv*, vol. abs/1512.03012, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:2554264

[7] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[8] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, and O. Hilliges, "D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands," in *ECCV*, 2022.

[10] H. Zhang, Y. Ye, T. Shiratori, and T. Komura, "Manipnet: Neural manipulation synthesis with a hand-object spatial representation," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–14, 2021.

[11] K. Karunratanakul, J. Yang, Y. Zhang, M. Black, K. Muandet, and S. Tang, "Grasping field: Learning implicit representations for human grasps," in *2020 International Conference on 3D Vision (3DV)*, Nov. 2020.

[12] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "Contactdb: Analyzing and predicting grasp contact via thermal imaging," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8709–8719.

[13] Y. Hasson, G. Varol, I. Laptev, and C. Schmid, "Towards unconstrained joint hand-object reconstruction from rgb videos," *2021 International Conference on 3D Vision (3DV)*, pp. 659–668, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237091509

[14] P. Tendulkar, D. Sur'is, and C. Vondrick, "Flex: Full-body grasping without full-body grasps," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21 179–21 189, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253761353

[15] H. Cheng, Y. Wang, and M. Q.-H. Meng, "Grasp pose detection from a single rgb image," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4686–4691.

[16] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, and M. Pollefeys, "H2o: Two hands manipulating objects for first person interaction recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 138–10 148.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[18] R. Li, J. Tanke, M. Vo, M. Zollhofer, J. Gall, A. Kanazawa, and C. Lassner, "Tava: Template-free animatable volumetric actors," 2022.

[19] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, pp. 1 – 14, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259267917

[20] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[21] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[22] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013. [Online]. Available: http://handtracker.mpi-inf.mpg.de/projects/handtracker_iccv2013/

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[24] G. Casiez, N. Roussel, and D. Vogel, "1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.

[25] P. Zuccotti, *Every Thing We Touch: A 24-hour Inventory of Our Lives*. Penguin UK, 2015.

[26] J. Z. Zheng, S. De La Rosa, and A. M. Dollar, "An investigation of grasp type and frequency in daily household and machine shop tasks," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4169–4175.

[27] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[28] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[29] J. Lundell, E. Corona, T. N. Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyrki, "Multi-fingan: Generative coarse-to-fine sampling of multi-finger grasps," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4495–4501.

[30] J. Lundell, F. Verdoja, and V. Kyrki, "Ddgc: Generative deep dexterous grasping in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6899–6906, 2021.

[31] Y. Ye and C. K. Liu, "Synthesis of detailed hand manipulations using contact sampling," *ACM Transactions on Graphics (ToG)*, vol. 31, no. 4, pp. 1–10, 2012.

[32] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*. Springer, 2022, pp. 201–221.

[33] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency

of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[34] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "Grab: A dataset of whole-body human grasping of objects," in *European conference on computer vision*. Springer, 2020, pp. 581–600.

[35] H. Jiang, S. Liu, J. Wang, and X. Wang, "Hand-object contact consistency reasoning for human grasps generation," in *Proceedings of the International Conference on Computer Vision*, 2021.

[36] J. DelPreto, C. Liu, Y. Luo, M. Foshey, Y. Li, A. Torralba, W. Matusik, and D. Rus, "Actionnet: A multimodal dataset for human activities using wearable sensors in a kitchen environment," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[37] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR 2019 - IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, United States: IEEE, Jun. 2019, pp. 11 799–11 808. [Online]. Available: https://hal.science/hal-02429093

[38] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik, "Reconstructing hand-object interactions in the wild," in *ICCV*, 2021.

[39] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[40] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 47–57, 2005.

[41] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[42] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *European Conference on Computer Vision*. Springer, 2016, pp. 294–310.

[43] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer, 2012, pp. 640–653.

[44] I. M. Bullock, T. Feix, and A. M. Dollar, "The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 251–255, 2015.

[45] G. Rogez, J. S. Supancic, and D. Ramanan, "Understanding everyday hands in action from rgb-d images," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3889–3897.

[46] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3196–3206.

[47] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[48] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/OccludedHands/

[49] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/

[50] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *European Conference on Computer Vision*. Springer, 2020, pp. 548–564.

[51] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 813–822.

[52] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, "Understanding human hands in contact at internet scale," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9869–9878.

[53] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "Contactpose: A dataset of grasps with object contact and hand pose," in *European Conference on Computer Vision*. Springer, 2020, pp. 361–378.

[54] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.

[55] H. Hamer, J. Gall, T. Weise, and L. Van Gool, "An object-dependent hand pose prior from sparse training data," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 671–678.

[56] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool, "Tracking a hand manipulating an object," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1475–1482.

[57] A. Tsoli and A. A. Argyros, "Joint 3d tracking of a deformable object in interaction with a hand," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 484–500.

[58] D. Tzionas and J. Gall, "3d object reconstruction from hand-object interactions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 729–737.

[59] J. Romero, H. Kjellström, and D. Kragic, "Hands in action: real-time 3d reconstruction of hands in interaction with objects," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 458–463.

[60] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2088–2095.

[61] T.-H. Pham, N. Kyriazis, A. A. Argyros, and A. Kheddar, "Hand-object contact force estimation from markerless visual tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2883–2896, 2017.

[62] T. H. E. Tse, Z. Zhang, K. I. Kim, A. Leonardis, F. Zheng, and H. J. Chang, "S^2contact: Graph-based network for 3d hand-object contact estimation with semi-supervised learning," in *ECCV*, 2022.

[63] H. Joo, T. Simon, and Y. Sheikh, "Total capture: A 3d deformation model for tracking faces, hands, and bodies," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8320–8329.

[64] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[66] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[67] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 65:1–65:14, Jul. 2019.

[68] V. Sitzmann, M. Zollhoefer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/b5dc4e5d9b495d0196f61d45b26ef33e-Paper.pdf

[69] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 2021. [Online]. Available: http://arxiv.org/abs/2106.10689v1

[70] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," *arXiv preprint arXiv:2112.05131*, 2021.

[71] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," *arXiv preprint arXiv:2203.09517*, 2022.

[72] L. Liu, M. Habermann, V. Rudnev, K. Sarkar, J. Gu, and C. Theobalt, "Neural actor: Neural free-view synthesis of human actors with pose control," *ACM Trans. Graph.(ACM SIGGRAPH Asia)*, 2021.

[73] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *CVPR*, 2021.

[74] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *ICCV*, 2021.

[75] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "HumanNeRF: Free-viewpoint rendering of moving people from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 210–16 220.

[76] E. Corona, T. Hodan, M. Vo, F. Moreno-Noguer, C. Sweeney, R. Newcombe, and L. Ma, "Lisa: Learning implicit shape and appearance of hands," in *CVPR*, 2022.

[77] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[78] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[79] S. Lin, A. Ryabtsev, S. Sengupta, B. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8758–8767, 2020.

[80] H. K. Cheng, J. Chung, Y.-W. Tai, and C.-K. Tang, "Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8887–8896, 2020.

[81] PhotoRoom, "Photoroom v5.0.9," 2023. [Online]. Available: https://www.photoroom.com/

[82] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[83] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "Humor: 3d human motion model for robust pose estimation," in *International Conference on Computer Vision (ICCV)*, 2021.

[84] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022.

[85] T. W. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "Dˆ2neRF: Self-supervised decoupling of dynamic and static objects from a monocular video," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=rG7HZZtIc-

[86] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conference on Computer Vision (ECCV)*, 2020.

[87] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[88] A. Mundra, J. Wang, M. Habermann, C. Theobalt, M. Elgharib *et al.*, "Livehand: Real-time and photorealistic neural hand rendering," *arXiv preprint arXiv:2302.07672*, 2023.

[89] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi, "Hoi4d: A 4d egocentric dataset for category-level human-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 013–21 022.

[90] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation," in *IEEE Computer Vision and Pattern Recognition Conference*, 2022.

[91] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura, "Patterns of static prehension in normal hands," *The American journal of occupational therapy*, vol. 34, no. 7, pp. 437–445, 1980.

[92] T. Li, M. Slavcheva, M. Zollhöfer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, and Z. Lv, "Neural 3d video synthesis," *CoRR*, vol. abs/2103.02597, 2021. [Online]. Available: https://arxiv.org/abs/2103.02597

[93] J. S. Yoon, K. Kim, O. Gallo, H. S. Park, and J. Kautz, "Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5336–5345.

[94] Z. Yan, C. Li, and G. H. Lee, "Nerf-ds: Neural radiance fields for dynamic specular objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8285–8295.

[95] F. Wang, S. Tan, X. Li, Z. Tian, Y. Song, and H. Liu, "Mixed neural voxels for fast multi-view video synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 706–19 716.

[96] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *CVPR*, 2023.

[97] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv preprint arXiv:2308.09713*, 2023.

[98] Y. Li, L. Zhang, Z. Qiu, Y. Jiang, N. Li, Y. Ma, Y. Zhang, L. Xu, and J. Yu, "Nimble: a non-rigid hand model with bones and muscles," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–16, 2022.

[99] N. Qian, J. Wang, F. Mueller, F. Bernard, V. Golyanik, and C. Theobalt, "Html: A parametric hand texture model for 3d hand reconstruction and personalization," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 54–71.

[100] K. Karunratanakul, S. Prokudin, O. Hilliges, and S. Tang, "Harp: Personalized hand reconstruction from a monocular rgb video," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 802–12 813, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254853916

[101] H. Zhang, S. Christen, Z. Fan, L. Zheng, J. Hwangbo, J. Song, and O. Hilliges, "Artigrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation," *arXiv preprint arXiv:2309.03891*, 2023.

[102] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and W. Xinggang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[103] X. Chen, T. Jiang, J. Song, M. Rietmann, A. Geiger, M. J. Black, and O. Hilliges, "Fast-snarf: A fast deformer for articulated neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[104] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[105] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.

[106] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[107] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[108] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[109] "Easymocap - make human motion capture easier." Github, 2021. [Online]. Available: https://github.com/zju3dv/EasyMocap

[110] U. Castiello, "The neuroscience of grasping," *Nature Reviews Neuroscience*, vol. 6, pp. 818–818, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:90493256

[111] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[112] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.