Visual-Tactile Sensing for In-Hand Object Reconstruction

Wenqiang Xu[*1,2], Zhenjun Yu[*1], Han Xue[1], Ruolin Ye[3], Siqiong Yao[1], Cewu Lu[§1,2]

[1]Shanghai Jiao Tong University   [2]Shanghai Qi Zhi institute   [3]Cornell University

[1]{vinjohn, jeffson-yu, xiaoxiaoxh, yaosiqiong, lucewu}@sjtu.edu.cn
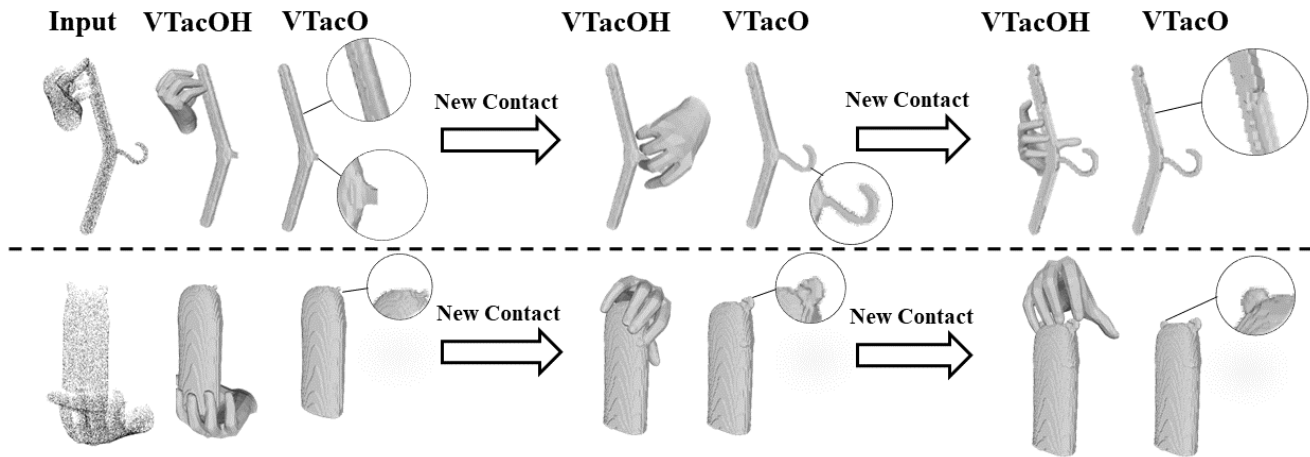
[3]ry273@cornell.edu

Figure 1. Our proposed visual-tactile learning framework **VTacO** and its extended version **VTacOH** can reconstruct both the rigid and non-rigid in-hand objects. It also supports refining the mesh in an incremental manner.

Abstract

Tactile sensing is one of the modalities humans rely on heavily to perceive the world. Working with vision, this modality refines local geometry structure, measures deformation at the contact area, and indicates the hand-object contact state. With the availability of open-source tactile sensors such as DIGIT, research on visual-tactile learning is becoming more accessible and reproducible. Leveraging this tactile sensor, we propose a novel visual-tactile in-hand object reconstruction framework VTacO, and extend it to VTacOH for hand-object reconstruction. Since our method can support both rigid and deformable object reconstruction, no existing benchmarks are proper for the goal. We propose a simulation environment, VT-Sim, which supports generating hand-object interaction for both rigid and deformable objects. With VT-Sim, we generate a large-scale training dataset and evaluate our method on it. Extensive experiments demonstrate that our proposed method can outperform the previous baseline methods qualitatively and quantitatively. Finally, we directly apply our model trained in simulation to various real-world test cases, which display qualitative results. Codes, models, simulation environment, and datasets are available at https://sites.google.com/view/vtaco/.

1. Introduction

Human beings have a sense of object geometry by seeing and touching, especially when the object is in manipulation and undergoes a large portion of occlusion, where visual information is not enough for the details of object geometry. In such cases, vision-based tactile sensing is a good supplement as a way of proximal perception. In the past, few vision-based tactile sensors were commercially available or open-source, so the visual-tactile sensing techniques could not be widely studied. Previous works [27, 34] on in-hand object reconstruction either studied rigid objects or were limited to simple objects with simple deformation.

---

* indicates equal contributions.

§ Cewu Lu is the corresponding author, the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China and Shanghai Qi Zhi institute.

Vision-based tactile sensors [6, 15, 30, 33] can produce colorful tactile images indicating local geometry and deformation in the contact areas. In this work, we mainly work with DIGIT [15] as it is open-source for manufacture and is easier to reproduce sensing modality. With tactile images, we propose a novel **V**isual-**Tac**tile in-hand **O**bject reconstruction framework named **VTacO**. VTacO reconstructs the object geometry with the input of a partial point cloud observation and several tactile images. The tactile and object features are extracted by neural networks and fused in the Winding Number Field (WNF) [13], and the object shape is extracted by Marching Cubes algorithm [17]. WNF can represent the object shape with open and thin structures. The poses of tactile sensors can be determined either by markers attached or by hand kinematics. By default, VTacO assumes the tactile sensor poses can be obtained independently, but we also discuss how to obtain the tactile sensor poses alongside the object with hand pose estimation. The corresponding method is named **VTacOH**.

With tactile information, we can enhance pure visual information from three aspects: (1) **Local geometry refinement**. We use tactile sensing as proximal perception to complement details of local geometry. (2) **Deformation at contact area**. Objects, even those we consider rigid, can undergo considerable deformation given external forces exerted by hand. (3) **Hand-object contact state**. Tactile sensors indicate whether the hand is in contact with the object's surface. To demonstrate such merits, we conduct the object reconstruction tasks in both rigid and non-rigid settings.

Since obtaining the ground truth of object deformation in the real world is hard, we first synthesize the training data from a simulator. DIGIT has an official simulation implementation, TACTO [29]. However, it is based on pybullet [5], which has limited ability to simulate deformable objects. Thus, we implement a tactile simulation environment **VT-Sim** in Unity. In VT-Sim, we generate hand poses with GraspIt! [18], and simulate the deformation around the contact area with an XPBD-based method. In the simulation, we can easily obtain depth image, tactile image, DIGIT pose, and object WNF as training samples for both rigid and non-rigid objects.

To evaluate the method, we compare the proposed visual-tactile models with its visual-only setting, and the previous baseline 3D Shape Reconstruction from Vision and Touch (3DVT) [23]. Extensive experiments show that our method can achieve both quantitative and qualitative improvements on baseline methods. Besides, since we make the tactile features fused with winding number prediction, we can procedurally gain finer geometry reconstruction results by incrementally contacting different areas of objects. It can be useful for robotics applications [8, 22]. Then, we directly apply the model trained with synthesis data to the real world. It shows great generalization ability.

We summarize our contributions as follows:

- A visual-tactile learning framework to reconstruct an object when it is being manipulated. We provide the object-only version VTacO, and the hand-object version VTacOH.

- A simulation environment, VT-Sim, which can generate training samples. We also validate the generalization ability of the models trained on the simulated data to the real-world data.

2. Related Works

Vision-based Tactile Sensors   Tactile sensors have been studied by robotics and material communities for decades. As a result, different approaches to mimic human tactile sensing are developed. Among them, the most relevant approaches are the vision-based tactile sensors [6, 15, 30, 33]. These methods typically use a gel to contact the object, and a camera sensor to observe the deformation through the layer. If the gel is calibrated, the deformation corresponds with a force so that we can obtain the contact force by observing the tactile images. Among these sensors, DIGIT [15] is the one with the structure most suitable for the human hand. The fact that it is attached to fingers rather than wrapping around makes it aligned with human fingertip structures.

Vision-based tactile sensors, by nature, have the merits of high-density, multi-DOF force sensing and a data format of image, which is suitable for neural network processing. These features make them beneficial for the computer vision community. However, it could be hard to re-implement most tactile sensors without open-source circuit design and drive programs. These essential components are necessary for the development of visual-tactile research. As an open-source hardware platform, DIGIT [15] takes the first step toward addressing this bottleneck problem.

Visual-Tactile Sensing for Object Reconstruction   Visual information usually provides global and coarse shape information in object reconstruction with visual-tactile sensing, while tactile information refines local geometry. [23, 25, 31].

Among these works, [23] is most relevant to our method. It jointly processes visual and tactile images to estimate global and local shapes. However, it is limited to watertight objects due to its SDF-based shape modeling approach. Besides this limitation, previous works [8, 25, 31] are also limited to static object settings. To our knowledge, none of the previous works considers hand-object interaction and thus ignores the deformation during manipulation.

In-hand Object Reconstruction   Earlier works usually consider interacting with rigid [19] or articulated objects

[28]. Most of them require prior knowledge about the objects. [27] studied jointly tracking a deformable object and hand during interaction, with required knowledge about object model and texture.

With the development of deep learning recently, the learning algorithms [4, 10, 11, 26] now have the potential to generalize to unseen objects. Among them, InteractionFusion [34] which also studies the deformation during manipulation is the most relevant one to our method. However, it adds a requirement for tracking the object from its rest state to model contact deformation.

3. Method

3.1. Overview

We consider the problem of 3D reconstruction of in-hand objects from a visual-tactile perspective, with which we propose **VTacO**, a novel deep learning-based visual-tactile learning framework.

In this setting, we assume the visual input is a 3D point cloud $\mathcal{P}$ of an in-hand object converted from a depth image. The tactile input $\mathcal{T} = \{T_i\}_{i=1}^N$ contains N tuples of tactile readings obtained from the sensors, where $T_i = \{(I_i, \boldsymbol{p}_i)\}$, I_i is the RGB tactile image, and $\boldsymbol{p}_i$ is the 6DOF sensor pose.

The object shape is modeled by the winding number field (WNF), a variant of the signed distance field (SDF). Thus, we can also learn the object representation similar to the SDF-based pipelines [20]. The point cloud $\mathcal{P} \in \mathbb{R}^{N_p \times 3}$ is converted to a global feature $\mathcal{F}_p \in \mathbb{R}^{d_p}$ through a feature extractor $\boldsymbol{f}_p(\cdot)$, where N_p is point number and d_p is the feature dimension. The tactile image I_i is forwarded to another feature extractor $\boldsymbol{f}_t(\cdot)$ to obtain local feature $\mathcal{F}_t \in \mathbb{R}^{d_t}$, where d_t is the feature dimension. Then, we sample M positions $\mathcal{X} = \{\boldsymbol{X}_j = (x_j, y_j, z_j)\}_{j=1}^M$ in the 3D space, and fuse $\mathcal{F}_p$ to each position, $\mathcal{F}_t$ to the near positions according to the sensor pose $\boldsymbol{p}$. The final feature will be passed to a pointwise decoder to predict the winding number value, which will be used for extracting the shape by Marching Cubes algorithm. The details will be discussed in Sec. 3.2

Considering the DIGIT sensor does not provide the sensor pose $\boldsymbol{p}$, we provide two different options to obtain $\boldsymbol{p}$ in the real world, namely by attaching markers or inferring from hand kinematics. For the latter option, we propose a pipeline **VTacOH** to jointly estimate the object shape, hand, and sensor pose. It will be discussed in Sec. 3.3.2.

An overview of our approach is illustrated in Fig 3.

3.2. Visual-Tactile Object Reconstruction, VTacO

Winding Number Field, WNF We use WNF to model the object, it is proposed by [13] which can implicitly represent a certain surface of an object. Unlike how SDF models space occupancy, WNF tracks the solid angles of the surface. Generally, given a Lipschitz surface S and a reference
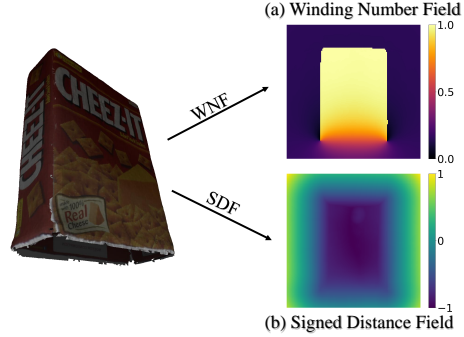

Figure 2. Object in **WNF** and **SDF** representation. For an object, the on-body surface is near 1, the off-body area is near 0 and the opening region is around 0.5. While SDF considers only the inside (< 0) and outside (> 0).

point $p \in \mathbb{R}^3$, if p is inside S, the winding number equals 1, and 0 if p outside. The winding number can also be used to model open, non-manifold surfaces and thus can model fine local geometry like thin structures, holes *etc*. More details of WNF can be referred to [13]. The difference between WNF and SDF is illustrated in Fig. 2. However, the explicit surface can also be extracted by Marching Cubes algorithm from WNF.

Visual Feature Encoder. Given the input point cloud $\mathcal{P}$, we adopt a 3D UNet-based volume encoder as the feature extractor $\boldsymbol{f}_p(\cdot)$. After six fully connected layers, the point cloud features go through a 3D-UNet [3] with four layers of both downsampling and upsampling, and we acquire the global feature $\mathcal{F}_p \in \mathbb{R}^{d_p}$ for every position in the space. We discuss different visual feature encoder choices in Sec. 5.5.

Tactile Feature Encoder. We extract the feature $\mathcal{F}_{t_i} \in \mathbb{R}^{d_t}$ from the tactile images I_i with $\boldsymbol{f}_t(\cdot)$. Though we can use different backbones for $\boldsymbol{f}_t(\cdot)$, in practice, we adopt ResNet-18 [12].

Position Sampling & Feature Fusion. Then, we sample M positions in 3D space to query the WNF state later through the decoder. We first try to find the positions relevant to tactile observations.

For each sensor i, we assume its sensor pose $\boldsymbol{p}_i$ is known individually. As we are aware that the tactile sensor like DIGIT itself cannot measure its pose in the real world, we will discuss the solution later (Sec. 3.3). Then, we estimate the depth of every pixel in each sensor's observation I_i using a simple 2D UNet $U_I(\cdot)$ with 3-layer CNN as an encoder. The relationship between the tactile pixel and the depth is associated with the gel deformation, which can be considered as a simple linear mapping. The physics behind it will be discussed in supplementary materials. Thus, the estimated depth represents contact geometry. We transform it from tactile sensor space to visual sensor space. In this way, we have M_1 points in 3D space which are relevant to
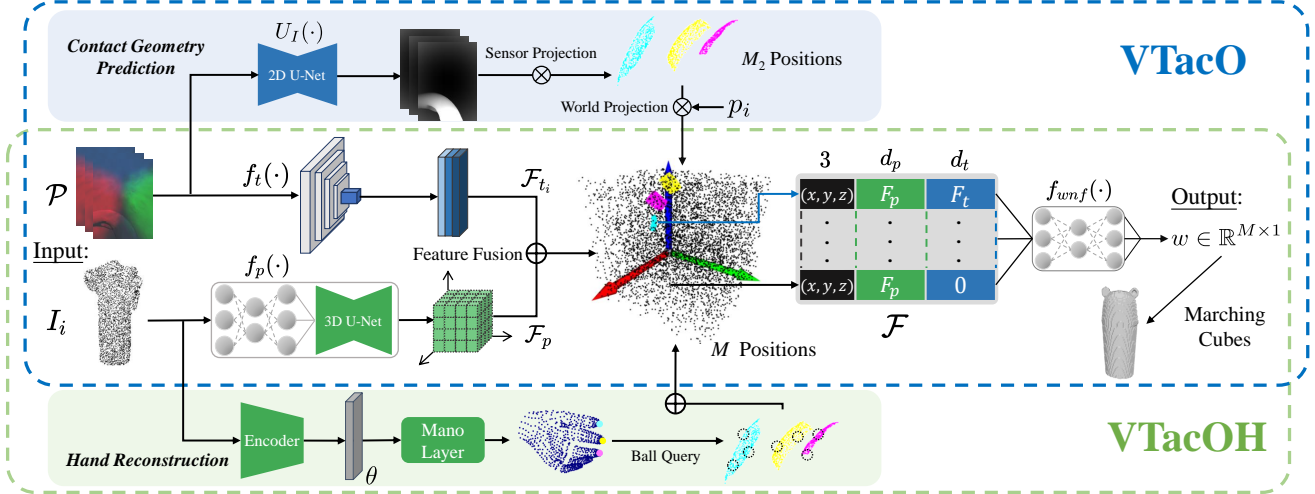
Figure 3. **VTacO** and **VTacOH**. The input is a set of point clouds and tactile images obtained by tactile sensors. We first compute the global point cloud information with a volume encoder and generate a local feature with a tactile feature encoder. For the position sampling, in **VTacO**, we predict the local point cloud with a U-Net from the tactile signals and project them to the visual camera space. And in **VTacOH** we reconstruct the hand by encoding MANO parameters and query positions around the local point clouds generated from the sensors on hand tips. For M sampled positions we first fuse the global and local features, then we estimate the winding number with the WNF decoder of several MLPs. For object reconstruction, we conduct marching cubes algorithm.

tactile images.

Then, we uniformly sample M_2 points in the 3D space, together with the M_1 positions, giving us $M = M_1 + M_2$ positions to query.

For each position in M_1, we fuse the location X_j with visual feature $\mathcal{F}_p$, and tactile feature $\mathcal{F}_t$. And in M_2, we substitute the tactile feature with all-0 vectors. We discuss different fusion strategies in Sec. 5.5. After fusing, we have position-wise feature $\mathcal{F} \in \mathbb{R}^d$, where d is the dimension of the fused features.

WNF Decoder. Finally, we use a 5-layer MLP $f_{wnf}(\cdot)$ as a decoder to predict the winding number w for each query, with $w = f_{wnf}(\mathcal{F})$.

Loss Function We train the whole pipeline in two parts.

First, we train U_I with pixel-wise L_1 loss. We can obtain the ground truth of the depth images for tactile sensors in simulation, but cannot obtain them in the real world. Thus, when dealing with real-world images, we directly apply the model learned in the simulation.

Then, we jointly train $\boldsymbol{f}_p$, $\boldsymbol{f}_t$, $\boldsymbol{f}_{wnf}$ with L_1 loss to supervise the winding number.

$$\mathcal{L}_w = |w - w^*|. \tag{1}$$

3.3. Sensor Pose Estimation

As aforementioned, the DIGIT sensor cannot measure its pose in the real world. To address this, we provide two options: attaching markers and inferring from hand kinematics.

3.3.1 Estimating Sensor Pose By Markers

Following the practice in pose estimation, we print ArUco tags [9] and attach them to fingertips. Then, we estimate the fingertip poses by estimating the pose of tags. We adopt OpenCV [1] to implement the pose estimation. It is a common practice, thus we leave the details in the supplementary materials.
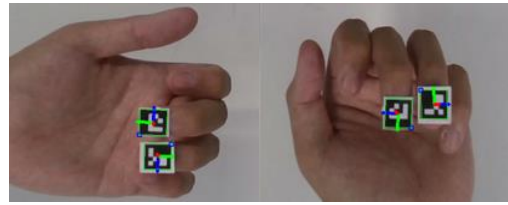

Figure 4. Estimating the fingertips' pose from the attached ArUco markers.

3.3.2 Estimating Sensor Pose With Hand Kinematics

Since the input $\mathcal{P}$ contains a view of the in-hand object, it allows us to reconstruct the hand alongside the object. We can infer the poses of tactile sensors based on the prior that they are attached to the fingertips. We can infer the poses with hand kinematics. To jointly learn visual-tactile hand-object reconstruction, we introduce **VTacOH**.

Hand-Sensor Pose Estimation. To reconstruct the hand, we adopt the parametric MANO model [21] which represents the hand shape with $\beta \in \mathbb{R}^{10}$, and the pose with

$\theta \in \mathbb{R}^{51}$, which contains the 6DOF wrist joint position and rotation and axis-angle rotation for 15 joints. Based on the MANO model, we regard a tactile sensor as a fixed extension adhered to the tip. The relative pose between the sensors and tips should remain constant. By recording the local transformation between the sensors and the tip positions, we can easily obtain the sensor pose after having the positions of the tips $\hat{J}_h^{tip} \in \mathbb{R}^{5\times3}$ through a differentiable MANO layer with θ and β.

Suppose the hand tips positions $\hat{J}_h^{tip}$ are predicted, then we have the translation $t_h \in \mathbb{R}^{5\times3}$ from the hand tips to the DIGIT sensor, giving us predicted DIGIT positions $\hat{p} = \hat{J}_h^{tip} + t_h$

Tactile Position Sampling with Hand Prediction. It is known that hand pose prediction can be inaccurate due to the error in neural network regression, which also influences the accuracy of the tactile sensor poses. Therefore, the tactile position sampling mentioned in Sec. 3.2 cannot be directly applied. Here, we present an easy but effective way to handle such circumstances. For every inferred DIGIT position $\hat{p}_i$, we simply query the sphere centered in the local point clouds generated from the tactile sensors with a radius $r = 0.1$. The feature fusion remains the same.

Loss Function In VTacOH, we will add additional supervision for hand reconstruction. It follows the common practice of MANO hand estimation by constraining the prediction of axis-angle of the hand joints with L_2 loss and we obtain:

$$\mathcal{L}_{hand} = ||\theta - \theta^*||_2^2 \qquad (2)$$

4. VT-Sim

We adopt a sim-to-real transfer paradigm to learn the visual-tactile learning task since obtaining information about object deformation in the real world is hard. And recently, many simulation environments have proved to be effective on softbody object simulation with Unity [7, 32].

We present **VT-Sim**, a simulation environment based on Unity to produce training samples of hand-object interaction with ground truths of WNF, visual depth images, tactile signals, and sensor poses effectively.

The dataset synthesis process is divided into two parts: **Hand-pose Acquisition** and **Interaction Simulation**.

Hand-pose Acquisition We use GraspIt! [18] and a 20-DOF hand model from GraspIt! assets to generate the grasp poses for the selected object models, and save the configuration of hand and objects after the grasp search is over. Since the MANO hand is slightly different from the hand we used to obtain the poses, we use a simple retargeting approach: We calculate the tip positions from the saved configuration of the 20-DOF hand model by forward kinematics. Then we align the tips between the 20-DOF hand and the tips of
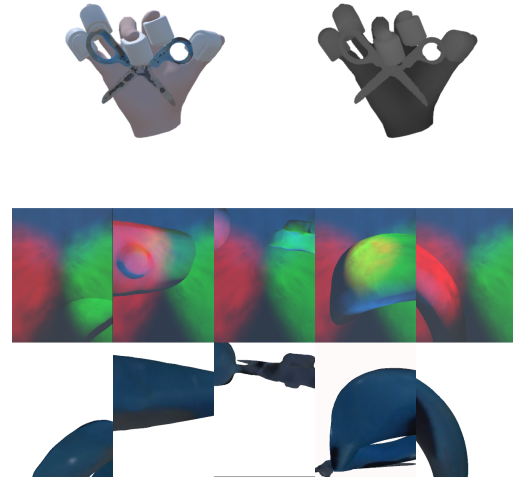

Figure 5. **VT-sim** example. The DIGIT sensors are attached to the finger tips. When grasping the object, the sensor gel applies forces and collides with the surface mesh, and the sensor camera captures images. The above demonstrates the RGB-D images from a certain view and the difference between tactile images with or without the light in the sensor.

the MANO hand with sensors and apply inverse kinematics to obtain the poses.

Interaction Simulation After obtaining the poses of MANO hands, sensors, and objects, we replay the configurations in Unity with collision. For the rigid object, we add the Rigidbody property and a mesh collider to it. For the soft body object, we model them with Obi [24], an XPBD-based physics engine. When we move the MANO hand with sensors to the retargeted pose, the hand sensor will collide with the object and forms the grasp. If the object is soft around the contact area, the contact region will be deformed.

Recording Setup We transplant the implementation of the DIGIT camera from TACTO [29] to obtain tactile images. Besides, we set visual cameras on 8 different viewpoints to generate RGB-D images. The point clouds can be converted from the RGB-D images with camera poses. We utilize the library igl [14] to generate the winding number field from the object meshes.

Finally, through one VT-Sim step, we obtain a full data point for training, which includes input point cloud $\mathcal{P}$, tactile signals $\mathcal{T} = \{T_i\}_{i=1}^N$ for every tactile sensor, hand poses θ^* and winding number $w(x)$ for every $x \in \mathbb{R}^3$.

5. Experiments

5.1. Implementation

We set $N_p = 3000$ as the input point cloud $\mathcal{P}$ converted from the depth image, and randomly sample 100,000 posi-

tions in the space with 20,000 on the surface of the object, and subsample to $M = 2048$ for the WNF prediction. An RGB tactile image and depth obtained in VT-Sim have the size of 300×200. During the training, we set the dimension of global and local features $d_p = d_t = 32$. We set the resolution of voxelization from the visual feature encoder $D = 64$.

We first train the depth and DIGIT pose prediction part with batch size 12 and $LR = 1e - 4$ with Adam optimizer for 100 epochs and use the pre-trained parameters for VTacO training. For the training of VTacO, we set the batch size as 6 and $LR = 2e - 4$ with Adam optimizer to train on the whole simulated dataset for 100 epochs, and finetune on every type of object with batch size 4 and $LR = 1e - 4$ for 50 to 100 epochs according to different types. For the VTacOH model, we train the hand reconstruction in 100 epochs with batch size 16 and $LR = 1e - 4$ with Adam optimizer, then apply the model parameters on feature embedding and train the whole VTacOH with the same hyperparameters and process as VTacO. The overall time for training VTacO and VTacOH on RTX TITAN XP are around 24 hours.

5.2. Datasets

We use two object model benchmarks which provide full model meshes and texture for our reconstruction task. **YCB Objects Models** YCB benchmark [2] contains 77 object models, with 600 RGBD images from different angles of cameras and their calibration information. For each example model, it includes textured meshes generated with both Poisson reconstruction and volumetric range image integration, a point cloud of the object generated by all the viewpoints. For our experiments, we select 6 object models in the class **Box** for generating a simulated dataset.

AKB-48 Objects Models AKB-48 benchmark [16] is a large-scale articulated object knowledge base that consists of 2,037 real-world 3D articulated object models of 48 categories. For every object, it provides textured models in both whole and part meshes, and a full point cloud from all viewpoints. In both simulated and real experiments, we select 5 to 10 objects in the following 4 categories: **Bottle**, **Foldingrack**, **Lock**, **Scissor**.

Simulated Dataset With VT-Sim, we can generate adequate datasets based on two object model benchmarks. We consider the categories **Bottle** and **Box** as deformable objects with stiffness 0.2 and 0.3, and others as rigid. For every object, we first generate 100 hand poses from **Hand-pose Acquisition** in VT-Sim, and select 10 to 40 poses that can generate at least one successful touch from five fingertips' Digits. With 8 different views from the RGB-D camera in Unity, we generate 8 point cloud sets from the depth image for every successful grasp. The overall dataset contains more than 10000 data points from 5 categories of object models, with every category, varying from 1000 to 3000.

We split the dataset into 7500 for training, 1500 for testing, and 1000 for validation and visualization.

5.3. Metrics

We adopt several metrics to evaluate our method quantitatively: (1) Intersection over Union (IoU) over object volumes. (2) Chamfer Distance (CD), and (3) Earth Mover's Distance (EMD) between predicted vertices of mesh and the ground truth;

5.4. Results

We conduct plenty of experiments to show the capability of VTacO. With the above metrics, we compare our results with the visual-tactile baseline 3D Shape Reconstruction from Vision and Touch (denoted "3DVT") [23]. Meanwhile, since "3DVT" didn't study the reconstruction of deformable objects, we only compare the 3 non-deformable categories: **Foldingrack**, **Lock**, **Scissor**. We also report the object reconstruction result for VTacOH setting. Due to the accurate hand pose estimation, the results of VTacOH are only slightly worse than the VTacO where the sensor poses are the ground truth. The results are shown in Tab. 1. We can see that our method outperforms others in most categories in the metrics. Qualitative results are demonstrated in Fig. 6. As illustrated in both quantitative and qualitative results, VTacO has better reconstruction results. We speculate that our outperformance is first due to the point-wise approach we conducted, and WNF being a continuous function concerning positions in the space. More importantly, the Feature fusion method can better demonstrate the contact area and the subtle texture of the surfaces. In Sec. 5.5 we will elaborate more to demonstrate the significant effects after we introduce the tactile sensing, and to prove the effectiveness of our choice of encoder and feature fusion strategy.

5.5. Ablation Study

We conduct an ablation study on the next four aspects. We first compare the results between pure vision and the fusion of vision and tactile signals. By grasping the same object several times, we can procedurally obtain more refined structures without retraining. Finally, we study the influence brought by different choices of visual feature encoder and the feature fusion module. **Pure vision v.s. Visual-tactile.** We can see results in Fig. 6 and Tab. 1 that, with the introduction of tactile images, the overall performance of reconstruction is better. Particularly, for the deformable objects **bottles** and **boxes**, the local feature can complete the information occluded by the hand, and better demonstrate the deformation of the objects. For rigid objects, touches can collect features of subtle texture from the surface of small, thin objects. For instance, the category **Foldingrack** has been reconstructed much better since the hollow struc-

Metrics	Method	bottle	box	foldingrack	lock	scissor	mean
IoU	3D Vision and Touch	*	*	0.746	0.468	0.691	0.622
	Ours (Vision only)	0.884	0.934	**0.837**	0.877	0.763	0.857
	Ours (VTacO)	**0.887**	**0.950**	0.782	**0.916**	**0.777**	**0.860**
	Ours (VTacOH)	0.886	0.947	0.765	0.911	0.772	0.855
CD	3D Vision and Touch	*	*	**0.242**	2.631	3.206	1.268
	Ours (Vision only)	1.109	0.459	1.579	1.140	7.549	1.472
	Ours (VTacO)	**0.936**	**0.305**	1.360	**0.932**	**0.894**	**0.798**
	Ours (VTacOH)	0.948	0.312	1.432	0.955	0.945	0.916
EMD	3D Vision and Touch	*	*	0.052	0.175	0.081	0.090
	Ours (Vision only)	**0.054**	0.028	**0.026**	0.083	0.110	0.051
	Ours (VTacO)	0.059	**0.018**	0.028	**0.012**	**0.052**	**0.028**
	Ours (VTacOH)	0.056	0.024	0.027	0.035	0.068	0.042

Table 1. Quantitative results on YCB and AKB Objects. This table illustrates a numerical comparison among 3DVT, pure vision methods, and our approach VTacO. We measure IoU (The larger the better), Chamfer Distance ($\times 100$, the smaller the better), and Earth Mover's Distance (The smaller the better). CD and EMD are measured at 2048 points. * indicates that 3DVT has no clear settings for deformable objects: Bottle and Box.

ture has been discovered through DIGIT sensors, and the category **Lock** also has better performances in the upper area, after we touch and detect its upper mesh.

Procedural Tactile Embedding. Our method allows us to procedurally gather the tactile embeddings and predict better reconstruction results without retraining the network. As shown in Fig. 1, after gradually touching the object several times, the reconstruction result becomes better and better. For the rigid objects, the textures and structures become clearer when conducting more grasp, such as the gap structure of the folding rack. For deformable objects such as **Bottles**, the object geometry will be changed due to the deformation caused by the contact. Thanks to the contact geometry prediction, we can first recover the deformed geometry to its original shape and then reasons the structures to be completed. To demonstrate the deformable regions are indeed reconstructed, we also illustrate the unrecovered results in supplementary materials. Such property can also be used for reconstructing plastic object with sustainable deformation.

Visual Feature Encoder Choices. We mainly compare two types of Visual Feature Encoder: **Multi-Plane Encoder** and **Volume Encoder**. **Multi-Plane Encoder** mainly project the input point cloud onto three orthogonal planes (i.e., plane xy, xz, or yz), and by using 2D UNet to encode the plane features and adding three features together, we obtain the multi-plane feature $\mathcal{F}_p^{plane} \in \mathbb{R}^{d_p}$ as our global feature for sample positions.

The quantitative result in Tab. 2 shows that the Volume Encoder outperforms in all metrics of all categories. This is mainly because the global feature is considered 3D information and fused with the local feature point-wisely, and the voxelization process fits the prediction of WNF, which indicates the position of each voxel relative to the surface of the object.

Metrics	Encoder	bottle	box	foldingrack
IoU	Multi-Plane	0.851	0.945	0.767
	Volume	**0.887**	**0.950**	**0.782**
CD	Multi-Plane	1.296	0.732	**0.940**
	Volume	**0.936**	**0.305**	1.360
EMD	Multi-Plane	0.052	0.030	0.029
	Volume	0.059	**0.018**	**0.028**

Table 2. Measured metrics with **Multi-plane** and **Volume encoder**. The Volume encoder performs better in most categories.

Fusion Strategies. We conduct experiments on two simple kinds of global and local feature fusion strategies: **concatenation** and **addition**. With concatenation, the final fused feature dimension $d = 3 + 64$, while $d = 3 + 32$ for the addition operation. Results in Tab. 3 show that the method **addition** has a much better performance compared to the others. We speculate that since we mark all-0 local feature vectors for the sample positions that are not near the sensor, the addition can correctly combine the two kinds of feature, while the fusion feature after the concatenation process would contain too many zeros, making it hard to predict the WNF for the decoders.

5.6. Real-world Experiment

We validate the performance of learned VTacO in the real-world data. We select 3 unseen **Box**, and 3 unseen **Bottle** to demonstrate the deformable object reconstruction. We also select 3 unseen **Lock** categories to demonstrate the
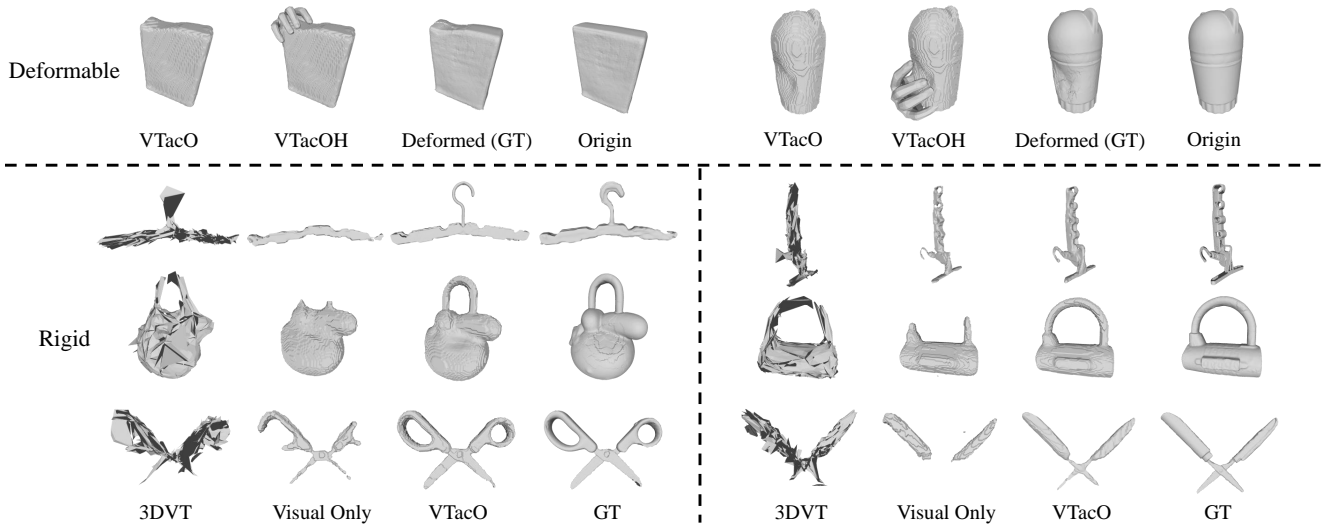
Figure 6. Qualitative example on **3DVT**, **Pure vision method**, **VTacO** and **VTacOH**. The upper demonstration shows our good performance on deformable objects. The results lower compare the visualization with different methods. We can clearly see that our method has a better qualitative reconstruction effect. In comparison with pure vision-based approaches and VTacO, we prove that the introduction of tactile signals improves the results significantly.

Metrics	Fusion	bottle
IoU	Addition	**0.9518**
	Concat	0.9339
CD	Addition	0.305
	Concat	**0.114**
EMD	Addition	**0.018**
	Concat	0.028

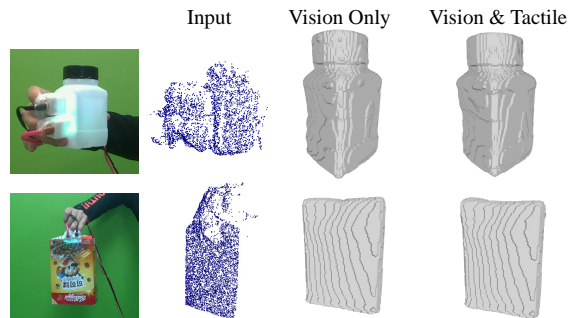Table 3. Quantitative results with different fusion strategies.

Figure 7. Real world examples. The deformation of the box and bottle has been better reconstructed with the introduction of tactile sensors in comparison with the pure-vision strategy.

procedural rigid reconstruction on real-world data. To note, the selected objects are not used in training from simulation. We adopt an Intel realsense L515 camera mounted on a fixed tripod to capture the depth observation. In the real world, we attach two DIGIT sensors to the hand and detect the sensor poses according to markers attached to the tips. Since we observe noise and shadows in real sensor readings, we augment the tactile images in simulation to match real RGB distribution by adding noise and adjusting contrast. We filter out the point cloud around the in-hand object manually and use it as input. In Fig. 7, we illustrate examples for **Box** and **Bottle** category, as for the **Lock** category, we demonstrate them in supplementary materials.

6. Conclusion and Future Works

In this work, we propose a novel visual-tactile in-hand object reconstruction framework, VTacO, and its extended version, VTacOH. Our proposed frameworks can reconstruct both rigid and non-rigid objects. The visual-tactile fusion strategy allows the framework to reconstruct the object geometry details incrementally. In the future, we are interested in leverage such features in robotics experiments to better estimate the object geometry in manipulation.

Acknowledgement

References

[1] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 4

[2] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 6

[3] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 3

[4] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5031–5041, 2020. 3

[5] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021. 2

[6] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018. 2

[7] Haoyuan Fu, Wenqiang Xu, Han Xue, Huinan Yang, Ruolin Ye, Yongxi Huang, Zhendong Xue, Yanfeng Wang, and Cewu Lu. Rfuniverse: A physics-based action-centric interactive environment for everyday household tasks. *CoRR*, abs/2202.00199, 2022. 5

[8] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10598–10608, 2022. 2

[9] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. 4

[10] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent 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*, pages 3196–3206, 2020. 3

[11] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[13] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. 2, 3

[14] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 5

[15] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 2

[16] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Yang Han, and Cewu Lu. Akb-48: A real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6

[17] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2

[18] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004. 2, 5

[19] Paschalis Panteleris, Nikolaos Kyriazis, and Antonis A Argyros. 3d tracking of human hands in interaction with unknown objects. In *BMVC*, pages 123–1, 2015. 2

[20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 3

[21] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017. 4

[22] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021. 2

[23] Edward Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdzal. 3d shape reconstruction from vision and touch. *Advances in Neural Information Processing Systems*, 33:14193–14206, 2020. 2, 6

[24] Virtual Method Studio. 5

[25] Sudharshan Suresh, Zilin Si, Joshua G Mangelson, Wenzhen Yuan, and Michael Kaess. Shapemap 3-d: Efficient shape mapping through dense touch and vision. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7073–7080. IEEE, 2022. 2

[26] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European conference on computer vision*, pages 581–600. Springer, 2020. 3

[27] Aggeliki Tsoli and Antonis A Argyros. Joint 3d tracking of a deformable object in interaction with a hand. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018. 1, 3

[28] Dimitrios Tzionas and Juergen Gall. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 729–737, 2015. 3

[29] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, 2022. 2, 5

[30] Shaoxiong Wang, Yu She, Branden Romero, and Edward Adelson. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6468–6475. IEEE, 2021. 2

[31] Shaoxiong Wang, Jiajun Wu, Xingyuan Sun, Wenzhen Yuan, William T Freeman, Joshua B Tenenbaum, and Edward H Adelson. 3d shape perception from monocular vision, touch, and shape priors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1606–1613. IEEE, 2018. 2

[32] Ruolin Ye, Wenqiang Xu, Haoyuan Fu, Rajat Kumar Jenamani, Vy Nguyen, Cewu Lu, Katherine Dimitropoulou, and Tapomayukh Bhattacharjee. Rcareworld: A human-centric simulation world for caregiving robots. *arXiv preprint arXiv:2210.10821*, 2022. 5

[33] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 2

[34] Hao Zhang, Zi-Hao Bo, Jun-Hai Yong, and Feng Xu. Interactionfusion: real-time reconstruction of hand poses and deformable objects in hand-object interactions. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019. 1, 3