End-to-end Surface Reconstruction For Touching Trajectories

Jiarui Liu[1,2], Yuanpei Zhang[1], Zhuojun Zou[1], and Jie Hao[1,2]✉

[1] Institute of Automation, Chinese Academy of Sciences
[2] Guangdong Institute of Artificial Intelligence and Advanced Computing

Abstract. Whereas vision based 3D reconstruction strategies have progressed substantially with the abundance of visual data and emerging machine-learning tools, there are as yet no equivalent work or datasets with which to probe the use of the touching information. Unlike vision data organized in regularly arranged pixels or point clouds evenly distributed in space, touching trajectories are composed of continuous basic lines, which brings more sparsity and ambiguity. In this paper we address this problem by proposing the first end-to-end haptic reconstruction network, which takes any arbitrary touching trajectory as input, learns an implicit representation of the underling shape and outputs a watertight triangle surface. It is composed of three modules, namely trajectory feature extraction, 3D feature interpolation, as well as implicit surface validation. Our key insight is that formulating the haptic reconstruction process into an implicit surface learning problem not only brings the ability to reconstruct shapes, but also improves the fitting ability of the network in small datasets. To tackle the sparsity of the trajectories, we use a spatial gridding operator to assign features of touching trajectories into grids. A surface validation module is used to tackle the dilemma of computing resources and calculation accuracy. We also build the first touching trajectory dataset, formulating touching process under the guide of Gaussian Process. We demonstrate that our method performs favorably against other methods both in qualitive and quantitative way. Insights from the tactile signatures of the touching will aid the future design of virtual-reality and humanrobot interactions.

1 Introduction

Studying the mechanics of rebuilding the surface of objects through touching will complement vision-based scene understanding. Humans have the ability to imagine the shape of an unknown object by sliding along the surface, some of them are even capable of identifying among different persons. This is usually accomplished by first exploring the surfaces of target objects, then integrating the information from different locations and imagining the whole shape according to prior knowledge. Although 3D reconstruction has been a classical research topic in vision and robotics applications, reconstruction from touching is still an uncharted territory to explore. Here in this paper, we design a data-driven algorithm to learn an uniform 3D representation space with touching information, and generate detailed 3D surface mesh as output.

Touching trajectories are usually treated as continuous curves, which can be discretized as a list of 3D coordinates. As the exploration can be started at any position and the local situation of the exploration is different each time, one target object may result in countless touching coordinate sequences. If our reconstruction framework is carried out in the order of exploration, it will be difficult to learn features that truly represent the global shape of objects. Our key insight is that the information contained in each trajectory is independent to the exploring orderswe can treat a touching trajectory as a special kind of point cloud that is dense in some local direction but extremely sparse in a global way.

In the recent few years, many neural network-based methods have been pioneered to mine the information contained in disorganized data such as point clouds, namely Multi-Layer Perceptron based methods[24][25], voxelization based methods[3][10][15], and graph based methods[33][34]. [39] takes partial point clouds as input, extract features with gridding and 3D convolution function, and obtain the final completed point cloud by predicting the coordinates after performing reverse gridding and feature sampling. However, none of these methods is designed to deal with touching trajectories. Besides, most of these methods still take more operations for the final presentation of a shape, namely, a triangular mesh. Reconstructing or generating from low-information domain brings extra difficulties, especially when there is no bidirectional mapping between 3D shape domain and this domain. [41] and [18] learns geometric transformations between a low-information domain and the 3D surfaces domain without taking point-to-point correspondences, namely transforming between meso-skeletons and surfaces, partial and complete scans, etc. However, data from those two domains are required to follow a global one-to-one mapping relationship. In our situation, touching trajectories are generated in a more random way. There isn't a specific mapping function from a complete shape to a touching trajectory, on the contrary, each target shape corresponds to a continuous touching space.

To address the issues mentioned above, we introduce an end-to-end haptic reconstruction network[3], which takes a sparse point cloud as input and generates triangular meshes as output. The framework of our method is illustrated in Fig. 1. Instead of learning the coordinates of new generated point clouds, we learn an implicit manifold for each target shape and use a random spatial sampling mechanism to supervise the generated implicit surface. During inference stage, we reconstruct a watertight triangular surface under the same framework by organizing the sampled points as the vertices of high-resolution grids. We also build a customized touching trajectory dataset to support the training of our network. We formulate an ordinary touching logic into a Gaussian process(GP) learning problem, and use the variance as a guide to explore the surface to somewhere that has not been touched. We assume the mapping function from the implicit surface to the touched surface points as a Gaussian Process, the goal of exploring the surface as much as possible is thus simplified to exploring

[3] Dataset and source code can be found: `https://github.com/LiuLiuJerry/TouchNet`

towards high variances, which indicates a high uncertainty during touching. We demonstrate the effectiveness of our method in both quantitative and qualitive way. It has been proved that our method not only succeeds in accomplishing touching reconstruction task, but also outperforms other potential methods by a large margin.

Our contributions are listed as follows:

- To the best of our knowledge, we are the first to pioneer the reconstruction from touching information. Our touching-reconstruction method can be taken as the first step of haptic shape learning and multimodal understanding.
- We propose the first touching reconstruction network, which embeds irregular exploration paths into an uniform feature space and generates watertight triangle meshes as output.
- We present the first touching trajectory dataset to support more data-driven methods requiring touching trajectory data.
- This method will serve as a knowledge base for learning and rebuilding from very irregular point clouds.

As the first step of haptic-reconstruction, this framework can be used in haptic identification, 3D object creation and many other practical applications.

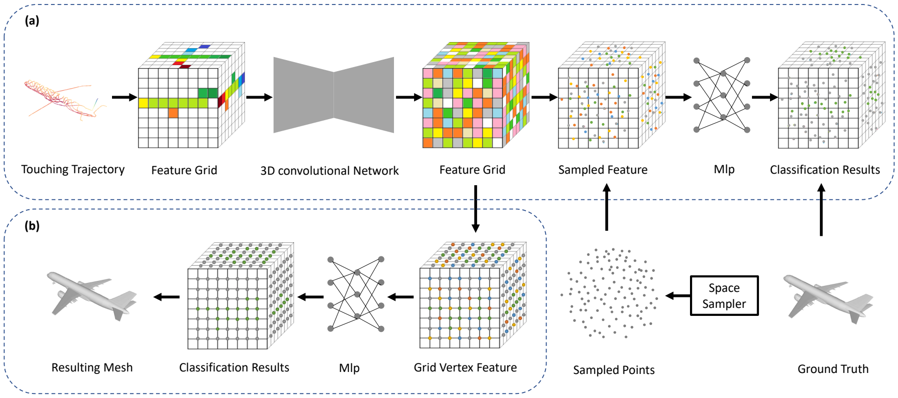

Fig. 1. Overview of our method. (a) Training process. (b) Reconstruction process.

2 Related Work

The existing relative research works involve both 3D reconstruction and haptic perception fields, which are detailed as follows.

3D Reconstruction. Existing 3D reconstruction algorithms are mostly designed to generate shapes from images or videos, while few of them generate from other domains such as contour lines, meso-skeletons or latent codes derived from a learned implicit manifold space. The internal mechanism can be further categorized into deformation-based methods and generation-based methods.

Deformation based methods assume the target shape is topological homeomorphism with a given shape, learning the deformation between the origin shape and the target shape. [17] optimizes object meshes for multi-view photometric consistency by posing it as a piecewise image alignment problem. [12] incorporates texture inference as prediction of an image, learning the deformable model with one single image. These kind of methods are widely used in human face and body reconstruction tasks, basing on the widely used 3D morphable face model 3DMM[6] and 3D morphable human shape model SMPL[19]. [8] first proposes to harness the power of Generative Adversarial Networks (GANs) to reconstruct the facial texture. [26] proposes a learning based method to learn complete 3D models with face identity geometry, albedo and expression parameters from images and videos. To avoid the statistical dependency contained in training dataset, [35] estimates 3D facial parameters using several independent networks, which results in a much bigger description space. In human shape reconstruction field, the problem tends to be more complicated with the disturbing information introduced by clothes. [1] learns to reconstruct the 3D human shape from few frames. It first encodes the images of the person into pose-invariant latent codes, then predicts the shape using both bottom-up and top-down streams.

To relax the reliance on the model's parameter space, [13] encodes the template mesh structure within the network from a single image using a GraphCNN. [21] constructs local geometry-aware features for octree vertices and designs a scalable reconstruction pipeline, which allows dividing and processing different parts in parallel for large-scale point clouds. [28] proposes to learn an implicit representation of each 3D shape. With the proposed end-to-end deep learning method, highly detailed clothed humans are digitized. The key insight of learning the implicit representation also inspired our work which has been presented in this paper. Other researchers predict a complete shape represented as point clouds, such as [20].

Besides generating from vision information, [16] proposes to learn generating semantic parts in a step by step way, and assembles them into a plausible structure. [18] augments 3D shapes by learning the generation process under the guidance of a latent manifold space. Utilizing the power of PointNet++ [24][25], [41] generates point surfaces from meso-skeletons by learning the displacement between two domains, without relying on point-to-point correspondences. However, this method still requires the shape of same domain organized in a uniform way, and the mapping is assumed to be bidirectional, which is not suitable in the touching procedural, while one shape will have countless touching results. [39] addresses the irregularity problems by proposing three differentiable layers: Gridding, Gridding Reverse, and Cubic Feature Sampling. The Gridding and Reverse Gridding operation has been proved to be an efficient way to learn both local and global features, but the generation framework is not capable of predicting smooth surface when the given point clouds are extremely sparse.

Haptic Perception. Tactile perception as well as tactile information based learning are commonly used in robot manipulation. For the purpose of object perception, plenty of work has been proposed to learn the feature of the local surface, such as [4,22,14,30,7,2,27,36,9]. Those work usually collects multimodal information using specially designed sensing components, using operation including pressing, sliding and tapping, and finally classifies the material or the shape of the objects using different mathematical methods, such as wavelet transform, dictionary learning, convolutional neural networks, etc. To interpretate tactile data in a global way, [29] proposes to treat tactile arrays obtained from different grasps using tactile glove as low-resolution single channel images, and utilizes the strong power of deep convolutional neural network to identify objects using filtered tactile frames. However, the grasping gestures as well as trajectories indicating spatial location information are not taken into consideration, which limits the perception of objects' 3D shape. [40] and [5] propose active learning frameworks based on optimal query paths to efficiently address the problem of tactile object shape exploration. Limited by the normal estimation algorithm as well as the flexibility of manipulators, those methods only work for objects with very primitive shapes, and the reconstruction results only indicate a rough outline of the actual objects. [32] and [31] incorporate depth and tactile information to create rich and accurate 3D models. However, the tactile information is only acquired as augmentation, while the majority of a target shape is still provided by depth camera. Here in our work, we demonstrate that using tactile information only has been sufficient for detailed high-quality shape reconstruction tasks.

3 Haptic Reconstruction Network

We treat haptic trajectory reconstruction as a particular shape completion problem, where the input data is extremely sparse in a global way, but is continuous along the exploration direction. Overall, it contains very limited information comparing to partial point clouds taken by a depth camara, and contains more ambiguity compared to contour lines or meso-skeletons. The size of the dataset is also limited by the big difficulty of collecting touching data, which challenges the ability of reconstruction network to learn the key feature of a "good shape" as well. Here we adapt the idea of gridding input touching trajectory points into a grid for feature extraction, and propose our own designment for the touching reconstruction task, including trajectory feature extraction, 3D feature interpolation and implicit surface validation.

3.1 Trajectory Feature Exaction

In recent years, many neural network-based methods have been pioneered to mine the information contained in unorganized data such as point clouds. One type of them is to use the Multi-Layer Perceptions and pooling functions to aggregate information across points. These kind of methods do not fully consider the connectivity across points and the context of neighboring points. Another

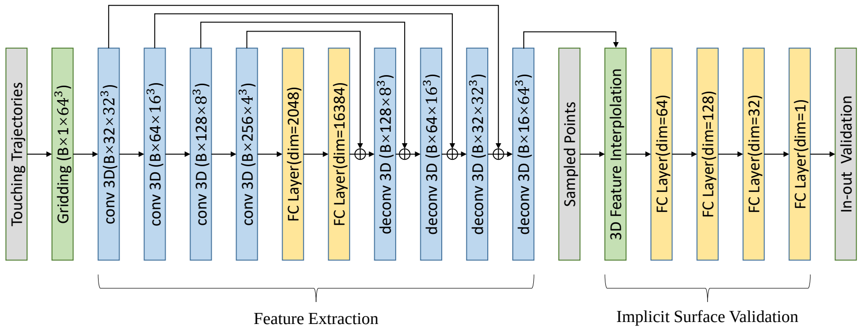

Fig. 2. Architecture of our touching reconstruction network. $\oplus$ denotes feature addition. The number of sampled points is set to 4000.

type of approaches are to voxelize the point cloud into binary voxels, then apply 3D convolutional neural networks to those voxels. Limited by the resolution of the voxels, theses methods have to drop the high-precision location information, resulting in a waste of the precision of the original data. Inspired by GRNet[39], here we apply gridding operation to explicitly preserve the structural and context information.

Taken a touching trajectory as inputs, gridding operator assigns the trajectory points into regular grids. For each point of the point cloud, the corresponding grid cell the point lies in is found according to their coordinates. Then eight vertices of the 3D grid cell are weighed using the function that explicitly measures the geometric relations of the point cloud. By applying this operation, the point clouds are mapped into 3D grids without loss of data accuracy. Then we adopt a 3D convolutional neural network with skip connections to extract the global features indicated by the sparse touching trajectories. As is detailed in Fig. 2, the architecture of our network follows the idea of U-net[38]. It has four 3D convolutional layers, two fully connected layers and four transposed convolutional layers. Each convolutional layer has a bank of 4^3 filters with padding of 2, while each transposed convolutional layer has a bank of 4^3 filters with padding of 1 and stride of 2. The numbers of output channels of two fully connected layers are 2048 and 16384. Features are flattened into one-dimensional data before being taken into fully connected layers. Batch normalization and leaky ReLU activation are used to prevent the network from gradient vanishing problem. After that, a sampler as well as Multilayer Perceptron Layers are used to transform the learned feature manifold into *in-out* labels of each sampled points, which is detailed in Section 3.3.

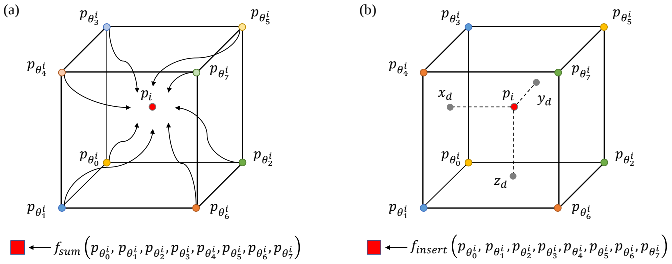

Fig. 3. Different feature aggregation methods. (a)Sum operation. (b)Trilinear interpolation.

3.2 3D Feature Interpolation

Feature aggregation, namely feature fusion and feature sampling, has been an important subject in neural network related researching. Common feature fusion operations are meanly accomplished by global concatenation and addition. Addition and concatenation of eight vertices of the 3D grid cell where each point lies in has been proved to aggregate the local context information of each point and build a connection between 3D gridding features and sampled points. However, this function does not take into account the relative distances between each point and the corresponding vertices of grid cell, which play important roles in traditional geometry descriptor computation.

We introduce a new feature aggregation operator, called 3D Feature Interpolation, to integrate features of surrounding grid vertices to sampled points. This operation assumes that the gridding features are a regular discretization of a continuous feature manifold. A manifold is a topological space which has the property of Euclidean space locally, every local area of which is linear and homeomorphic to the Euclidean space. Built on this assumption, we perform interpolation based on the distance to the grid cell edges it lies in. Trilinear interpolation, widely known as a intuitive and explicable interpolation method in graphics, is chosen as our feature interpolation operator.

Let $\mathcal{S} = \{f_1^v, f_2^v, ..., f_{t^3}^v\}$ be the feature map of 3D CNN, where t^3 is the size of the feature map, and $f_i^v \in \mathbb{R}^c$. Given a sampled point p_i, its corresponding feature is signed as f_i^c, which is computed as

$$\begin{aligned} f_i^c = {} & f^c_{\theta_0^i}(1-x_d)(1-y_d)(1-z_d)+ \\ & f^c_{\theta_1^i}x_d(1-y_d)(1-z_d)+ \\ & f^c_{\theta_2^i}(1-x_d)y_d(1-z_d)+ \\ & f^c_{\theta_3^i}(1-x_d)(1-y_d)z_d+ \\ & f^c_{\theta_4^i}x_d(1-y_d)z_d+ \\ & f^c_{\theta_5^i}(1-x_d)y_d z_d+ \\ & f^c_{\theta_6^i}x_d y_d(1-z_d)+ \\ & f^c_{\theta_7^i}x_d y_d z_d, \end{aligned} \tag{1}$$

where x_d, y_d, and z_d is the proportional distance to corresponding cell edges:

$$\begin{cases} x_d = D(p_i, p_{\theta_0^i})/D(p_{\theta_1^i}, p_{\theta_0^i}) \\ y_d = D(p_i, p_{\theta_1^i})/D(p_{\theta_2^i}, p_{\theta_1^i}) \\ z_d = D(p_i, p_{\theta_2^i})/D(p_{\theta_3^i}, p_{\theta_2^i}). \end{cases} \tag{2}$$

$\{p_{\theta_j^i}\}_{j=1}^8$ denotes the coordinate of the vertices the point p_i lies in, $\{f_{\theta_j^i}\}_{j=1}^8$ denotes the features of eight vertices, and $D(,)$ computes the Euclidean distance between two coordinates. Note that this operator is simple and differentiable, which makes it easy to be extended to other tasks.

3.3 Implicit Surface Validation

Representing target objects as implicit surfaces, the formulation of the ground truth shape is written as:

$$\mathcal{S} = \left\{x \in \mathbb{R}^3 | \mathcal{F}(x) = 1\right\}, \tag{3}$$

where $\mathcal{S} \in \mathbb{R}^3$ is the one level set of function $\mathcal{F} : \Omega \in \mathbb{R}^3 \to \mathbb{R}$. The touching trajectories are denoted as set of point clouds $\mathcal{P} = \{P_i\}_{i=1}^w$.

To validate the implicit surface learned by our network, we use a rejection-sampling strategy to quantify the error between implicit surfaces and ground truth shapes. To strengthen the ability of the network to learn the details, we adapt an Gaussian distribution based adaptive sampler. It first samples points on the surface of ground truth objects, then adds random displacements to the sampled points under the guide of a Gaussian distribution. After that, we sample uniformly inside the whole space and randomly choose 4000 points as the sampling results.

The validation process is mathematically modeled as a Multi-Layer Perception function. The numbers of output channels of fully-connected layer are designed as 64, 128, 32, 1. Finally, an *in-out* label is computed using a sigmoid function. Compared to generating point cloud positions, our method can learn a continuous implicit surface which can be transformed into triangular meshes for further usage. Besides, this rejection sampling method turns the complex position regression problem into a simpler two-way classification problem, which is more suitable for a network to excel.

During reference, the validation module is integrated into an octree based marching cubes algorithm. During reconstruction, the generation space is discretized and the vertices are organized in an octree structure for different resolutions. The inside-outside values are first calculated in a low resolution, and then the cubes whose vertex values vary from each other are evaluated in a higher resolution. We find the isosurface by calculating the position coordinates of each intersection point.

3.4 Loss Function

For the convenience of calculation, we denote the ground truth label as :

$$S_{gt} = \begin{cases} 1, & p \in mesh \\ 0, & p \notin mesh \end{cases} \tag{4}$$

The loss function is defined as the L2 distance between the predicted label and the ground truth:

$$L = \sum_{i \in P} \left\| S_{pred}^{i} - S_{gt}^{i} \right\|_2^2 \tag{5}$$

Here P is the point set sampled around the target shapes, S_{pred}^{i} is the ith label predicted by our network, and S_{gt}^{i} is the corresponding label indicating the *in-out* relationship with the ground truth shape.

4 Dataset Acquisition

Different from traditional shape reconstruction task, one challenge we handled is, there's no such a dataset for us to validate our algorithm directly. Here we propose our own haptic dataset, and introduce our method for the acquisition. Acquisition of the dataset containing the trajectory generated by real hands requires specially designed hardware and is a waste of time. Here we formulate the touching procedural as a process of minimizing the uncertainty of the target object under the guide of Gaussian Process, and build the touching dataset in computer simulation.

4.1 Gaussian Process

Gaussian process has been proposed as a regression method for implicit surfaces[40]. It not only approximates the underlying surface but also provides an uncertainty measure for the mesh. A GP models the probability $P(F(x)|\mathcal{P})$ as a Gaussian distribution, where $P(F(x)|\mathcal{P})$ is the probability of the implicit surface function F conditioned on the tactile data:

$$\mu_F(X) = m + \kappa(x)^T b \tag{6}$$

the variance is computed as:

$$\mathbb{V}_F(\mathrm{x}) = \ k(\mathrm{x}, \mathrm{x}) - \kappa(\mathrm{x})^T \mathrm{G}^{-1} \kappa(\mathrm{x}) \tag{7}$$

where κ, K and G represent matrixes of kernel functon k.

$\kappa(x) = (k(x; x_1), ..., k(x; x_w))^T$, $\mathrm{G} = \mathrm{K} + \sigma_n^2 \mathrm{I}_w \in \mathbb{R}^{w \times w}$, $\mathrm{K} = (k(x_i; x_j))_{i,j=1}^{w}$, $\mathrm{b} = \mathrm{G}^{-1}(\mathrm{Y} - m) \in \mathrm{R}^w$, $\mathrm{Y} = (c_1, ..., c_w)^T$. We set $m = 1$ as the prior mean, representing that most part of the space is empty.

For each time, we compute the contact points as well as the local normal at contact position, which will be used during exploration stage.

4.2 Touching Simulation

Aside from obtaining the surface from GP, at each time, our final goal is to compute a direction in the local Euclidean space and simulate the exploration result for the next time step. As what has been mentioned, we estimate the GP variance function as an evaluation for the uncertainty. Our assumption is that moving towards a most uncertain position will maximize the information obtained during the sliding and thus minimize the exploration time. We compute the gradient of the GP variance function as:

$$\mathrm{g}_{\mathbb{V}_F}(\mathrm{x}) = \frac{\partial k}{\partial \mathrm{x}}(\mathrm{x}, \mathrm{x}) - 2\kappa(\mathrm{x})^T \mathrm{G}^{-1} \frac{\partial \kappa}{\partial \mathrm{x}}(\mathrm{x}) \tag{8}$$

We donate the exploration direction as $\mathrm{g} = \frac{\mathrm{g}_{\mathbb{V}_F}(\mathrm{x})}{|\mathrm{g}_{\mathbb{V}_F}(\mathrm{x})|}$. As shown in 4(a), the exploration direction g could point at any direction in 3D space. To make this direction meaningful, we project g on the tangent space of target shape at point P_i, and normalize the projected $V_i^{'}$ as V_i. The next point P_{i+1} is obtained by moving point P_i towards direction V_i and projecting the moved point $P_{i+1}^{'}$ on the surface, noted as P_{i+1}.

Exploring objects of complex shape in this method directly might cause local optimal resultsthe exploring point moves inside a low variance region back and forth, ignoring other unexplored regions. This is because the exploring point walks into a region surrounded by places which have been explored before, thus the point might find the variance low enough to quit the exploration. In this situation the generated path does not contain a global information of target object, discriminative information might be missed. To avoid this, we stop the local marching and choose a new start position where the variance is estimated to be the highest. We stop exploring once the exploration exceeds certain steps and use the explored paths represented as points as input.

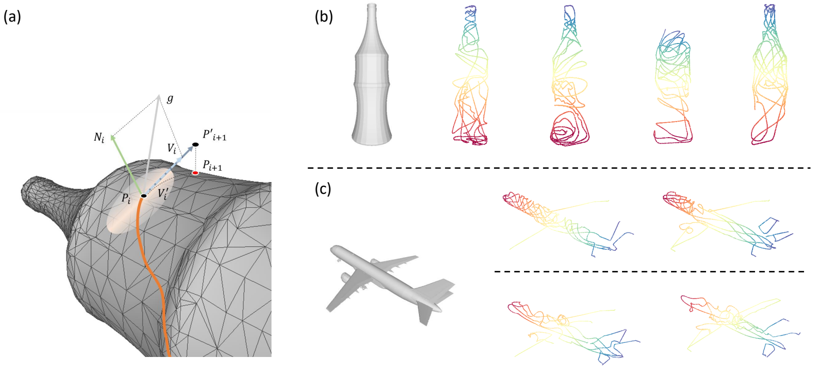

Fig. 4. Local trajectory planning and examples of touching simulation results. (a)Schematic diagram of local exploration direction calculation. (b)Touching simulation results of a bottle. (c)Touching simulation results of an airplane model.

5 Experiments

5.1 Implementation Details

We build the touching dataset from a subset of ShapeNet[37]. Before conducting touching simulation, we convert the triangle soups into watertight manifold surfaces using projection-based optimization method[11]. We organize three subjects of shapes, Bottle, Airplane, Car, which contains 106, 1972, 1989 target shapes respectively. For each shape, we simulate 4 touching trajectories with randomly chosen start points.

We implement our network using PyTorch[23] with Adam optimizer, and train the network with a batch size of 4 for 200 epochs on NVIDIA V100 GPUs. We use a multi-step learning rate and the initial learning rate is set to $1e-4$.

5.2 Quantitative Results

We conduct our evaluation on sampled points labeled as *in* for comparison. Let $T = (x_i, y_i, z_i)_{i=1}^{n_\mathcal{T}}$ be the ground truth and $S = (x_i, y_i, z_i)_{i=1}^{n_\mathcal{S}}$ be a reconstructed shape being evaluated, where $n_\mathcal{T}$ and $n_\mathcal{S}$ are the numbers of points of T and S, we use Chamfer Distance and F-Score as quantitative evaluation metrics.

Chamfer's distance. Following some existing generation methods[39][41], we take this as a metric to evaluate the global distance between two point clouds. Given two point clouds written as $\mathcal{S}$ and $\mathcal{T}$, this distance is defined as:

$$L_{CD} = \frac{1}{n_\mathcal{S}} \sum_{s \in \mathcal{S}} min_{t \in \mathcal{T}} \|s - t\|_2^2 + \frac{1}{n_\mathcal{T}} \sum_{t \in \mathcal{T}} min_{s \in \mathcal{S}} \|s - t\|_2^2 \tag{9}$$

Table 1. Evaluations of different reconstruction methods. We use $d = 1$ in F1-Score calculation. CD represents the Chamfer's distance multiplied by 10^{-3}. $-$ means that the network fails to learn a distribution and the resulting indicator is meaningless. Ours-FA represents our network with feature addition operator, Ours-FI represents our network with trilinear feature interpolation operator. The best results are highlighted in bold.

dataset	Indicators	PointNet++	GRNet	Ours-FA	Ours-FI
Bottle	F-score(@1%)	-	43.62%	90.08%	**92.74%**
	CD($\times 10^{-3}$)	-	0.44	0.2781	**0.2718**
Airplane	F-score(@1%)	-	62.28%	86.11%	**87.95%**
	CD($\times 10^{-3}$)	-	1.29	**0.2185**	0.2279
Car	F-score(@1%)	-	26.07%	91.24%	**91.37%**
	CD($\times 10^{-3}$)	-	0.7221	0.1234	**0.1203**

F1-score. To quantify the local distance between two shapes, we use F-score metric as:

$$L_{F-Score}(d) = 2\frac{PR}{P+R}, \tag{10}$$

where P and R are the precision and recall defined as the percentage of the points whose distance to the closest point in another shape(written as $min(,)$) is under threshold d:

$$P = \frac{1}{n_{\mathcal{S}}} \sum_{s \in \mathcal{R}} |min(s, \mathcal{T}) < d| \tag{11}$$

$$R = \frac{1}{n_{\mathcal{T}}} \sum_{t \in \mathcal{T}} |min(t, \mathcal{S}) < d| \tag{12}$$

Each metric mentioned above is evaluated on all these three dataset. We take the farthest point sampling method to sample the points into 2048-long points for alignment. Table 1 shows that our framework achieves good indicators even with an extremely small training set. In both cases, our method outperforms others by a large margin, indicating the excellence and effectiveness of our network.

5.3 Qualitive Results

The details of the ground truth shapes, simulated touching trajectories, results completed by GRNet, and the results of our methods are listed in Fig. 5. We demonstrate that our method can easily tackle the problem of tactile trajectory reconstruction. This method is compared with GRNet[39] and PointNet++-based displacement learning function[18]. The PointNet++-based algorithm can hardly learn the generation distribution of our irregular touching dataset. GRnet learn a rough shape of the target object, but the generated point clouds have blurred boundaries and lack detail. While sharing the gridding operation with GRNet, our implicit surface learning strategy outperforms coordinate generation strategy with more even points and smoother surfaces.

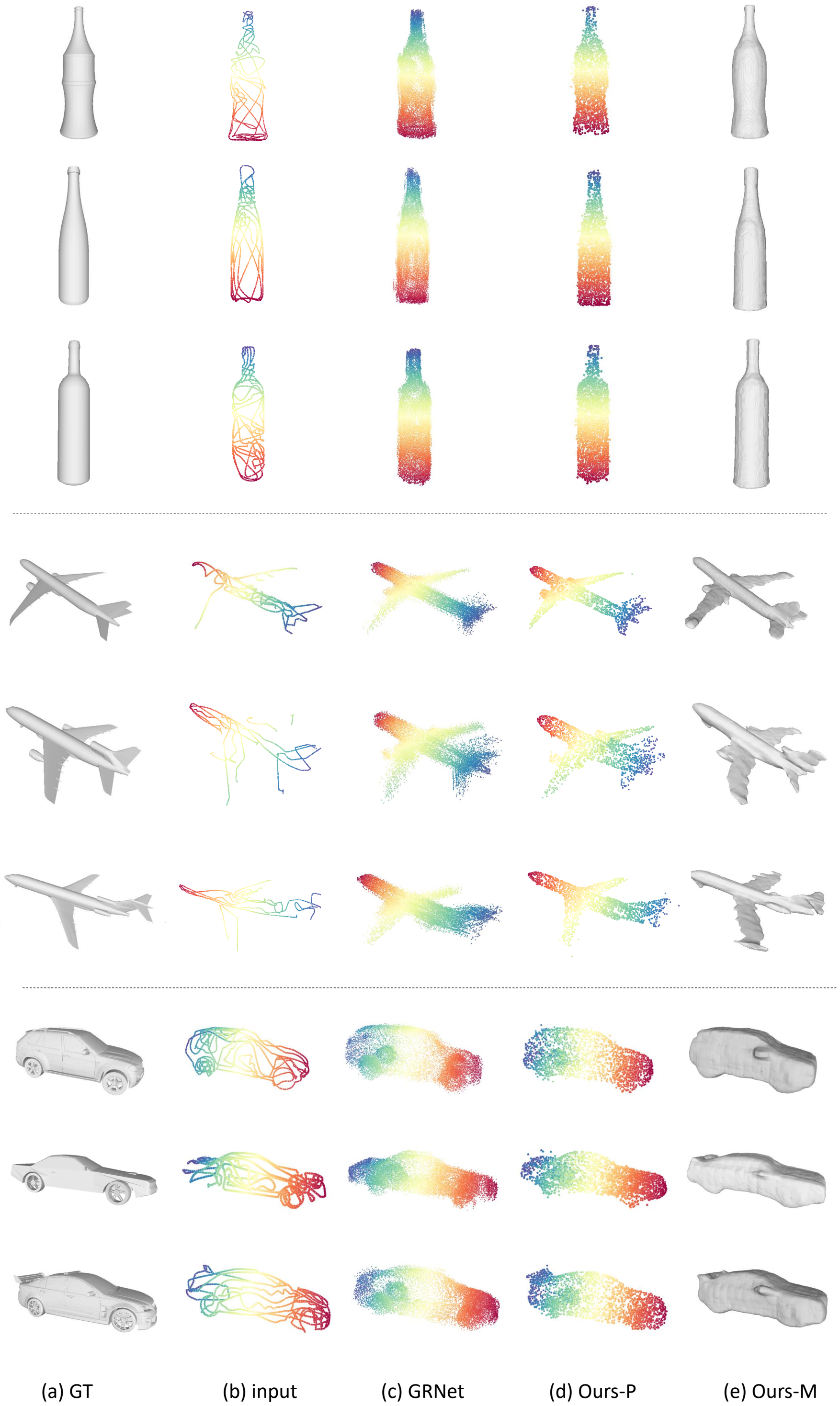

Fig. 5. Reconstruction results of Bottle and Airplane dataset. Column from left to right is: (a) Ground truth shape of target objects, (b) Touching trajectories generated from target objects, (c) Results from GRnet, (d) Sampled point clouds labeled as **in** by our network, (e) Surface mesh reconstructed using our method.

5.4 Ablation Study

To demonstrate the effectiveness of the 3D feature interpolation operator in the proposed method, evaluations are conducted on different feature integration methods, the results of which are also listed in Table 1. While network with feature addition operator works fine compared to other framework, it still lags behind the proposed network with feature interpolation operator, which confirms the hypothesis that taking relative distance between cube vertices and points into account can help networks learn a better distributions.

6 Conclusion

In this paper we study how to generate surface meshes directly from touching trajectories. The main motivation of this work is to treat the touching sequences as a specific kind of point clouds. To achieve this goal, we formulate the reconstruction problem as an implicit surface prediction task. We introduce our end-to-end reconstruction network, which contains trajectory feature extraction, 3D feature interpolation and implicit surface validation module. To the best of our knowledge, this network is the first method to reconstruct surface meshes from pure touching information. To validate the efficiency of our method, we also propose the first touching trajectory dataset. Both qualitive and quantitative evaluations as well as comparisons are conducted on the proposed dataset, which indicates the effectiveness and superiority of our method.

Acknowledgements This work is supported by the National Science and Technology Major Project from Minister of Science and Technology, China (Grant No. 2018AAA0103100), the Guangdong Provincial Key Research and Development Plan(Grant No. 2019B090917009), the Science and Technology Program of Guangzhou, China(Grant No. 202201000009).

References

1. Alldieck, T., Magnor, M., Bhatnagar, B.L., Theobalt, C., Pons-Moll, G.: Learning to reconstruct people in clothing from a single rgb camera. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1175–1186 (2019). https://doi.org/10.1109/CVPR.2019.00127
2. Chu, V., McMahon, I., Riano, L., McDonald, C.G., He, Q., Perez-Tejada, J.M., Arrigo, M., Darrell, T., Kuchenbecker, K.J.: Robotic learning of haptic adjectives through physical interaction. Robotics and Autonomous Systems **63**, 279–292 (2015)
3. Dai, A., Qi, C.R., NieSSner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6545–6554 (2017). https://doi.org/10.1109/CVPR.2017.693

4. Dallaire, P., Giguère, P., Émond, D., Chaib-Draa, B.: Autonomous tactile perception: A combined improved sensing and bayesian nonparametric approach. Robotics and autonomous systems **62**(4), 422–435 (2014)
5. Driess, D., Englert, P., Toussaint, M.: Active learning with query paths for tactile object shape exploration. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 65–72 (2017). https://doi.org/10.1109/IROS.2017.8202139
6. Egger, B., Smith, W.A.P., Tewari, A., Wuhrer, S., Zollhöfer, M., Beeler, T., Bernard, F., Bolkart, T., Kortylewski, A., Romdhani, S., Theobalt, C., Blanz, V., Vetter, T.: 3d morphable face models - past, present and future. CoRR **abs/1909.01815** (2019), http://arxiv.org/abs/1909.01815
7. Erickson, Z., Chernova, S., Kemp, C.C.: Semi-supervised haptic material recognition for robots using generative adversarial networks. In: Conference on Robot Learning. pp. 157–166. PMLR (2017)
8. Gecer, B., Ploumpis, S., Kotsia, I., Zafeiriou, S.: GANFIT: generative adversarial network fitting for high fidelity 3d face reconstruction. CoRR **abs/1902.05978** (2019), http://arxiv.org/abs/1902.05978
9. Giguere, P., Dudek, G.: A simple tactile probe for surface identification by mobile robots. IEEE Transactions on Robotics **27**(3), 534–544 (2011)
10. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 85–93 (2017). https://doi.org/10.1109/ICCV.2017.19
11. Huang, J., Zhou, Y., Guibas, L.: Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups. arXiv preprint arXiv:2005.11621 (2020)
12. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. CoRR **abs/1803.07549** (2018), http://arxiv.org/abs/1803.07549
13. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional mesh regression for single-image human shape reconstruction. In: CVPR (2019)
14. Kursun, O., Patooghy, A.: An embedded system for collection and real-time classification of a tactile dataset. IEEE Access **8**, 97462–97473 (2020)
15. Li, D., Shao, T., Wu, H., Zhou, K.: Shape completion from a single rgbd image. IEEE Transactions on Visualization and Computer Graphics **23**(7), 1809–1822 (2017). https://doi.org/10.1109/TVCG.2016.2553102
16. Li, J., Niu, C., Xu, K.: Learning part generation and assembly for structure-aware shape synthesis. CoRR **abs/1906.06693** (2019), http://arxiv.org/abs/1906.06693
17. Lin, C., Wang, O., Russell, B.C., Shechtman, E., Kim, V.G., Fisher, M., Lucey, S.: Photometric mesh optimization for video-aligned 3d object reconstruction. CoRR **abs/1903.08642** (2019), http://arxiv.org/abs/1903.08642
18. Liu, J., Xia, Q., Li, S., Hao, A., Qin, H.: Quantitative and flexible 3d shape dataset augmentation via latent space embedding and deformation learning. Computer Aided Geometric Design **71**, 63–76 (2019). https://doi.org/https://doi.org/10.1016/j.cagd.2019.04.017, https://www.sciencedirect.com/science/article/pii/S0167839619300330
19. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: a skinned multi-person linear model. ACM Trans. Graph. **34**, 248:1–248:16 (2015)

20. Mandikal, P., Babu, R.V.: Dense 3d point cloud reconstruction using a deep pyramid network. CoRR **abs/1901.08906** (2019), http://arxiv.org/abs/1901.08906
21. Mi, Z., Luo, Y., Tao, W.: Tsrnet: Scalable 3d surface reconstruction network for point clouds using tangent convolution. CoRR **abs/1911.07401** (2019), http://arxiv.org/abs/1911.07401
22. Oddo, C.M., Controzzi, M., Beccai, L., Cipriani, C., Carrozza, M.C.: Roughness encoding for discrimination of surfaces in artificial active-touch. IEEE Transactions on Robotics **27**(3), 522–533 (2011)
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. CoRR **abs/1912.01703** (2019), http://arxiv.org/abs/1912.01703
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. CoRR **abs/1612.00593** (2016), http://arxiv.org/abs/1612.00593
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. CoRR **abs/1706.02413** (2017), http://arxiv.org/abs/1706.02413
26. R., M.B., Tewari, A., Seidel, H., Elgharib, M., Theobalt, C.: Learning complete 3d morphable face models from images and videos. CoRR **abs/2010.01679** (2020), https://arxiv.org/abs/2010.01679
27. Richardson, B.A., Kuchenbecker, K.J.: Improving haptic adjective recognition with unsupervised feature learning. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3804–3810. IEEE (2019)
28. Saito, S., Huang, Z., Natsume, R., Morishima, S., Li, H., Kanazawa, A.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2304–2314 (2019). https://doi.org/10.1109/ICCV.2019.00239
29. Sundaram, S., Kellnhofer, P., Li, Y., Zhu, J.Y., Torralba, A., Matusik, W.: Learning the signatures of the human grasp using a scalable tactile glove. Nature **569**, 698–702 (05 2019). https://doi.org/10.1038/s41586-019-1234-z
30. Tulbure, A., Bäuml, B.: Superhuman performance in tactile material classification and differentiation with a flexible pressure-sensitive skin. In: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids). pp. 1–9. IEEE (2018)
31. Varley, J., DeChant, C., Richardson, A., Nair, A., Ruales, J., Allen, P.K.: Shape completion enabled robotic grasping. CoRR **abs/1609.08546** (2016), http://arxiv.org/abs/1609.08546
32. Varley, J., Watkins-Valls, D., Allen, P.K.: Multi-modal geometric learning for grasping and manipulation. CoRR **abs/1803.07671** (2018), http://arxiv.org/abs/1803.07671
33. Wang, K., Chen, K., Jia, K.: Deep cascade generation on point sets. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. p. 37263732. IJCAI'19, AAAI Press (2019)
34. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. CoRR **abs/1801.07829** (2018), http://arxiv.org/abs/1801.07829
35. Wen, Y., Liu, W., Raj, B., Singh, R.: Self-supervised 3d face reconstruction via conditional estimation. In: 2021 IEEE/CVF International Conference on Computer

Vision (ICCV). pp. 13269–13278 (2021). https://doi.org/10.1109/ICCV48922.2021.01304

36. Windau, J., Shen, W.M.: An inertia-based surface identification system. In: 2010 IEEE International Conference on Robotics and Automation. pp. 2330–2335. IEEE (2010)
37. Wu, Z., Song, S., Khosla, A., Tang, X., Xiao, J.: 3d shapenets for 2.5d object recognition and next-best-view prediction. CoRR **abs/1406.5670** (2014), http://arxiv.org/abs/1406.5670
38. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S., Tong, X.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. CoRR **abs/1901.11153** (2019), http://arxiv.org/abs/1901.11153
39. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: Gridding residual network for dense point cloud completion. CoRR **abs/2006.03761** (2020), https://arxiv.org/abs/2006.03761
40. Yi, Z., Calandra, R., Veiga, F., van Hoof, H., Hermans, T., Zhang, Y., Peters, J.: Active tactile object exploration with gaussian processes. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4925–4930 (2016). https://doi.org/10.1109/IROS.2016.7759723
41. Yin, K., Huang, H., Cohen-Or, D., Zhang, H.R.: P2P-NET: bidirectional point displacement network for shape transform. CoRR **abs/1803.09263** (2018), http://arxiv.org/abs/1803.09263