3D Shape Perception from Vision and Touch

by

Shaoxiong Wang

B.S. in Computer Science
Tsinghua University, 2017

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2019

Certified by .....
Edward H. Adelson
John and Dorothy Wilson Professor of Vision Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Elecrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

3D Shape Perception from Vision and Touch

by

Shaoxiong Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Perceiving accurate 3D object shape is important for robots to interact with the physical world. Current research along this direction has been primarily relying on visual observations. Vision, however useful, has inherent limitations due to occlusions and the 2D-3D ambiguities, especially for perception with a monocular camera. In contrast, touch gets precise local shape information, though its efficiency for reconstructing the entire shape could be low. In this thesis, we propose a novel paradigm that efficiently perceives accurate 3D object shape by incorporating visual and tactile observations, as well as prior knowledge of common object shapes learned from large-scale shape repositories. We use vision first, applying neural networks with learned shape priors to predict an object's 3D shape from a single-view color image. We then use tactile sensing to refine the shape; the robot actively touches the object regions where the visual prediction has high uncertainty. Our method efficiently builds the 3D shape of common objects from a color image and a small number of tactile explorations (around 10). Our setup is easy to apply and has potentials to help robots better perform grasping or manipulation tasks on real-world objects.

Thesis Supervisor: Edward H. Adelson
Title: John and Dorothy Wilson Professor of Vision Science

acknowledgments

I would like to first express my gratitude to Prof. Edward Adelson. It is my privilege to have him as my advisor. Ted is extremely knowledgeable and always shares wisdom us. More importantly, he has been very inspiring and encouraging. From him, I learned to aim for top-quality research, and how to refine the work step by step without haste.

I would like to say thank you to Prof. William Freeman for being a great academic advisor and sharing valuable experiences.

A special thank you to my best friend and roommate, Bai Liu, for all the supporting and discussion. He always encourage me when I was trapped by research problems. I also would like to say thank you to Dr. Wenzhen Yuan, from whom I have learned a great deal about how to be a good researcher. Her constructive suggestions and insightful ideas helped me so much.

I would thank my labmate for making the lab such a lovely group, Branden Romero, Sandra Liu, Achu Wilson, Filipe Veiga, and Yu She. I would also appreciate all the help and accompany from my friends, especially Siyuan Dong, Dongying Shen, Jiajun Wu, Xingyuan Sun, Zhengdong Zhang, Changchen Chen, Lei Xu, Yue Wang, Fengyi Li, Xiaoyue Gong, and Yilun Zhou.

Finally, I want to thank my parents for always believing in me and supporting me unconditionally in the past years.

Contents

1 Introduction **13**

2 Related Work **17**

2.1 3D Shape Completion from Vision 17

2.1.1 Matching from Database 17

2.1.2 Completion by Structure and Regularities 19

2.1.3 Gaussian Process Implicit Surface 20

2.1.4 Neural Network-based 21

2.2 Tactile Sensing for Shape Reconstruction 22

3 Method **25**

3.1 3D Reconstruction from Vision and Shape Priors 25

3.1.1 2.5D Sketch Estimation 27

3.1.2 3D Shape Estimation 27

3.2 Tactile Sensing for Shape Refinement 28

3.2.1 3D Reconstruction from GelSight 29

3.2.2 Registration of World and System Coordinates 29

3.2.3 Updating Shape Reconstruction with Touch 30

3.3 Policy for Active Tactile Exploration 32

4 Experiments **35**

4.1 Robotic System Setup 35

4.2 Conducting Touch without Collision 36

4.3 Dataset 37

4.4 Results 37

4.5 Shape Priors and the Exploration Policy 38
4.6 RGB vs RGB-D Input 40

5 Conclusion and Future Work **43**
5.1 Conclusion 43
5.2 Future Work 43
5.2.1 Efficiency 44
5.2.2 Immobilization 44
5.2.3 Evaluation on Robot Tasks 44

List of Figures

1-1 Our model for 3D shape reconstruction. It first reconstructs a rough 3D shape from a single-view color image, leveraging shape priors learned from large-scale 3D shape repositories. It then efficiently incorporates local tactile signals for shape refinement. 14

2-1 Matching from database [1] 18

2-2 Shape completion from different types of symmetry [2]. Blue: partial observation; Green: shape from symmetry. 19

2-3 Gaussian Process Implicit Surface representing (a) A simple "blob" defined by 15 points on the surface, one interior +1 point and 8 exterior -1 points arranged as a cube; (b) Two views of the Stanford bunny defined by 800 surface points, one interior +1 point, and a sphere of 80 exterior -1 points [3] 21

2-4 3D shape completion using Convolutional Neural Networks (CNN) and 3D shape synthesis for refinement [4] 22

2-5 (a) A cookie is pressed against the skin of an elastomer block. (b) The skin is distorted, as shown in this view from beneath. (c) The cookie's shape can be measured using photometric stereo and rendered at a novel viewpoint. [5] 23

3-1 An overview of our interactive system that estimates 3D shape from monocular vision, touch, and shape priors. 26

3-2 Our model has three major components. It first estimates the object's 2.5D sketches (depth, surface normals, and silhouette) from a single RGB image. It then recovers a rough 3D shape from them. Third, it integrates tactile signals to update the latent shape encoding and to generate a refined 3D shape. 27

3-3 Tactile signals on different parts of the object and the corresponding 3D reconstructions 28

3-4 Reprojection loss for touches. (a) When the sensor makes a touch attempt but fails to reach the object, the voxels along its trajectory should all be 0. (b) When the sensor contacts the object, the corresponding voxels should be 1, and all voxels in front of it along the trajectory should be 0. 31

3-5 Our policy on finding the next place to touch. (a) A 2D search grid overlaid on the voxel grid, where the confidence values of the voxel prediction are assigned to the search grid. (b) After the assignment, we compute the integral map and use it to efficiently search for the region of maximal uncertainty. See text for details. 32

4-1 Results on 3D shape perception. From a single RGB image, our model recovers a rough 3D shape using shape priors. The reconstruction often captures the basic geometry, but deviates from the actual shape in various ways. The results improve gradually with touch signals. For example, for the bell-shaped bottle in the last row, the initial reconstruction is too fat (best seen from the top-down view). With tactile signals, our model recovers its flat shape. Our system also corrects object pose, as shown in the water bottle case. 36

4-2 We show the effects of shape priors and the policy. If we Direct Edit the voxels' value (not using learned priors to update), each touch can only be used to update the shape locally. The shape does not change much even after many touches. With Random Policy, it takes longer for the model to obtain fine shape structure. 38

4-3 The two priors on the sugar box. A network trained on general shapes predicts a less accurate shape, which is later corrected by touches. A network trained on box-like shapes gives better results. 39
4-4 Shape estimation accuracy with respect to the number of touches, measured in Chamfer distance. Our policy recovers the shape accurately and efficiently. With Random Policy, it takes much longer to reconstruct a reasonable shape; if we Direct Edit the voxels' value (not using learned priors to update), the object is hardly updated after each touch. The Human method asks a human to manually select where to touch for each step and can be seen as a reference for comparison. 40
4-5 Our method can use either our estimated depth maps or Kinect depth maps. A Kinect depth map can be helpful if it is accurate: for example, the initial reconstruction of the left bottle is flatter using the Kinect depth map. However, if we purely rely on Kinect depth, our reconstruction would not be as accurate when the Kinect depth is inaccurate (see the transparent water bottle). 40

Chapter 1

Introduction

For a robot to effectively interact with the physical world, *e.g.*, to recognize, grasp, and manipulate objects, it is highly helpful to know the accurate 3D shape of the objects. 3D shape perception often relies on visual signals; however, using vision alone has fundamental limitations. For example, visual shape perception is often ambiguous due to the difficulties in discriminating the influence of reflection [6]; real-life occlusions and object self-occlusions also pose challenges to reconstruct full 3D shape from vision. The use of depth sensors alleviates some of these issues, though depth signals can also be too noisy to capture the exact object shape, and depth measurement is largely impacted by the object's color or transparency.

Touch is another way to perceive 3D shapes. The majority of tactile sensors measure the force distribution or geometry over a small contact area. A robot can use multiple touches, combined with the position and pose of the sensor in each touch, to reconstruct an object's shape without suffering from the ambiguity caused by its surface color or material [7]. Tactile sensing is however constrained by the size and scale of the sensor: as each touch only gets information of a local region, it may take many touches and a long time to reconstruct the full shape of an object.

A natural solution is to use tactile sensors to augment vision observations, just as human use fingers—using vision for rough shape reconstruction and touch exploration for shape refinement, especially in occluded regions. For example, Bjorkman *et al.* [8] explored refining visually perceived shape with touch, where they used a depth camera for a point cloud, a three-finger Schunk Dextrous hand for tactile data, and Gaussian

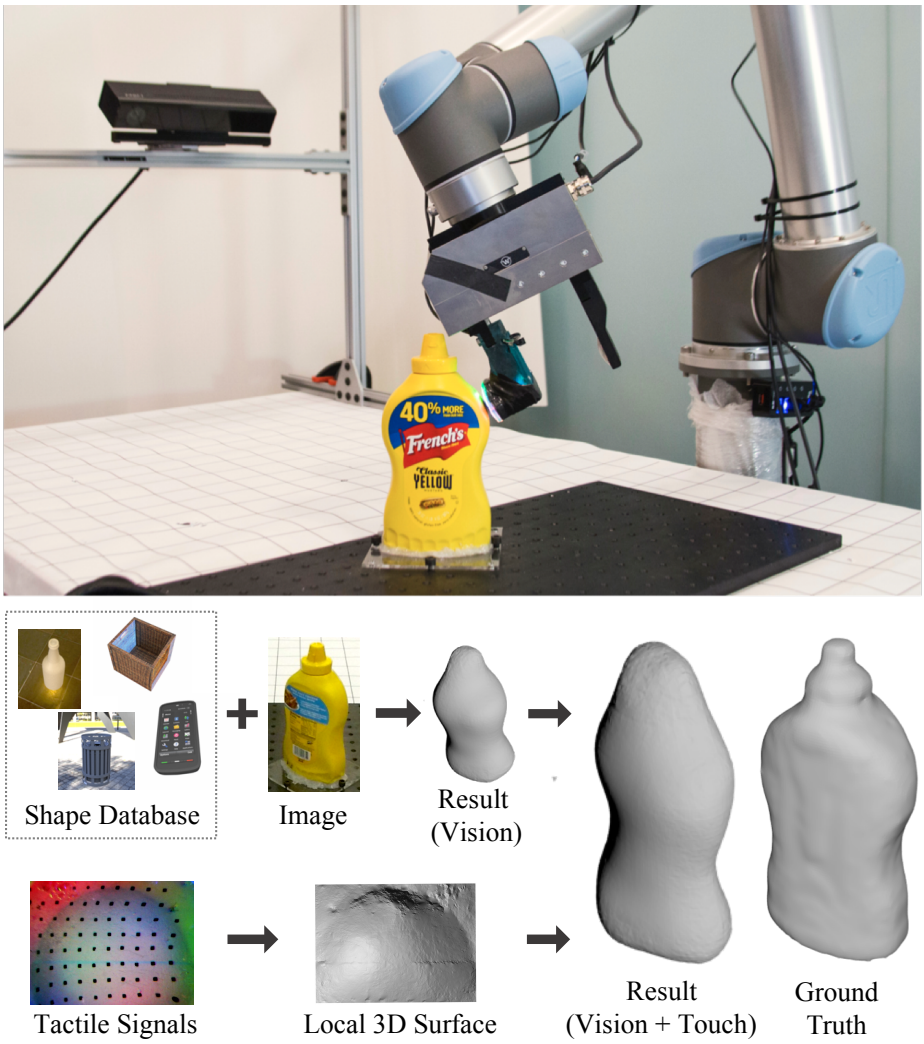


Figure 1-1: Our model for 3D shape reconstruction. It first reconstructs a rough 3D shape from a single-view color image, leveraging shape priors learned from large-scale 3D shape repositories. It then efficiently incorporates local tactile signals for shape refinement.

processes for shape prediction.

In this thesis, we propose a model that estimates the full 3D shape of common objects from monocular color vision, touch, and learned shape priors. We first use vision to predict the full 3D shape of the object from a monocular color and/or depth image, leveraging the power of 3D deep learning and large-scale 3D shape repositories. Specifically, our model is trained on many 3D CAD models and their RGB-D renderings; it learns to reconstruct a 3D shape from a color image by capturing implicit shape priors throughout the process. It generalizes well to real scenarios, producing plausible 3D shapes from a single image of real-world objects.

We then let the robot touch the object to refine the estimated shape. The tactile sensor we use is a GelSight sensor [9], which measures the geometry of local surface with high spatial resolution. By touching object surface with GelSight, the robot obtains additional constraints on the object geometry. Instead of making a local update to the reconstruction for each touch, which is inefficient, we incorporate local tactile constraints to refine the shape globally using the learned shape priors. Moreover, we propose an exploration policy that actively selects the touch point to maximally reduce the uncertainty in the shape prediction. This helps to reduce the number of touches needed.

We aim to make the system efficient and easy to apply. For efficiency, we use only one visual image and a few touch explorations (5–10 touches); for system simplicity, we use a fixed color camera and a tactile sensor on the effector of a 6-DOF robot arm. The setup can be easily applied to other robots as well.

We test our system on multiple common objects, and show that with a small number of touch exploration, the robot can predict the 3D object shape well. We also present ablation studies to qualitatively and quantitatively validate the effect of our learned shape priors and the active exploration policy. The system can be easily applied to other robots that have a high degree-of-freedom arm and an external color camera. This enables the robot to effectively perceive 3D object shape and to interact with the object.

Chapter 2

Related Work

2.1 3D Shape Completion from Vision

The problem of reconstruction of full 3D shapes from depth maps or partial scans can be defined as 3D shape completion. It has been widely studied in robotics, computer vision, and computer graphics. The input is partial observations of 3D objects, usually taken from depth cameras which tend to be noisy and have missing data. The goal is to predicate possible full 3D shapes from the observations. Compared to partial observations, the estimated full 3D shape can help robots perform better on tasks like grasp planning [10] [11].

Researchers explored different methods to solve this problem, including matches from the database, completion by structure and regularities, Gaussian process implicit surface and neural network based methods, etc. The following part gives an overview of different methods and discussion about their advantages and disadvantages.

2.1.1 Matching from Database

Since it's challenging to directly estimate the full 3D models, researchers explored to leverage the power of the database. The idea is to find the nearest neighbors in the database to represent the object given the partial scan. By this way, it's guaranteed to generate a real object from the database without hallucination.

The key techniques are how to separate different objects from a scene, and how to effectively retrieve the most similar 3D models from the database.

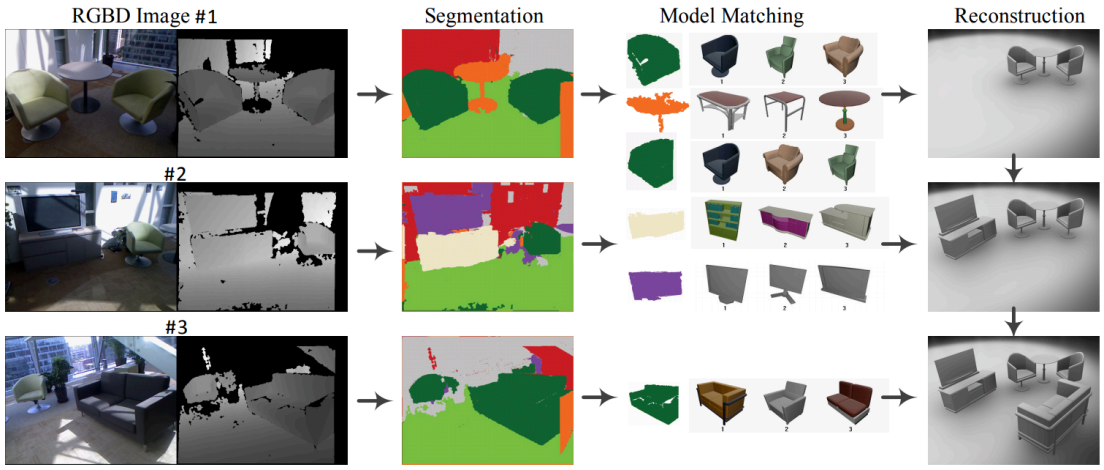


Figure 2-1: Matching from database [1]

Shao *et al.* [1] applied a conditional random field (CRF) to segment different objects. And they posed the model matching problem as a model instance recognition problem, trained a random forest on rendering depth images to model index in the database. The extracted features include depth difference, normal structure tensor, geometry moment and spin image. For pose estimation, they extended the random forest so that the children of each node also have different transformation distribution.

Nan *et al.* [12] proposed search-classify region growing method for segmentation, which trained a classifier first and iteratively expand the regions to maximize the prediction confidence. To fit the real data better, a non-rigid deformation based on iterative closest point (ICP) is applied on the template of each class. The features are based on the size ratio of the bounding box of different parts and angles between different parts.

Li *et al.* [13] extended classic 2D image stitching framework to 3D model matching. They first extract key points based on 3D Harris feature; then calculate the global and local descriptor for the key point based on signed distance function (SDF); finally, run random sample consensus (RANSAC) to match the 3D models.

In terms of evaluation, [1] [12] [13] mainly focused on running time, recognition accuracy and qualitative results.

The methods based on database guarantee to generate shapes that exist in the real world, and in many cases, the models from the database are good enough in tasks like scene understanding and navigation. The limitation is that it's difficult to

generalize to new objects if there are no similar objects in the database. And due to the hardware limitation of the depth sensors, only using depth information makes the methods difficult to perform well on small, thin, reflective, or transparent objects [13].

2.1.2 Completion by Structure and Regularities

Most of the man-made objects and many natural objects have symmetry and structures in them. In computer graphics, [14] showed that a lot of objects can be decomposed by a small number of parts with repetition, mirror, scale, and transformation. So when 3D scan missing some information, the structure and regularities can be used for 3D shape completion. The key technique is how to find symmetry and structures based on partial data.

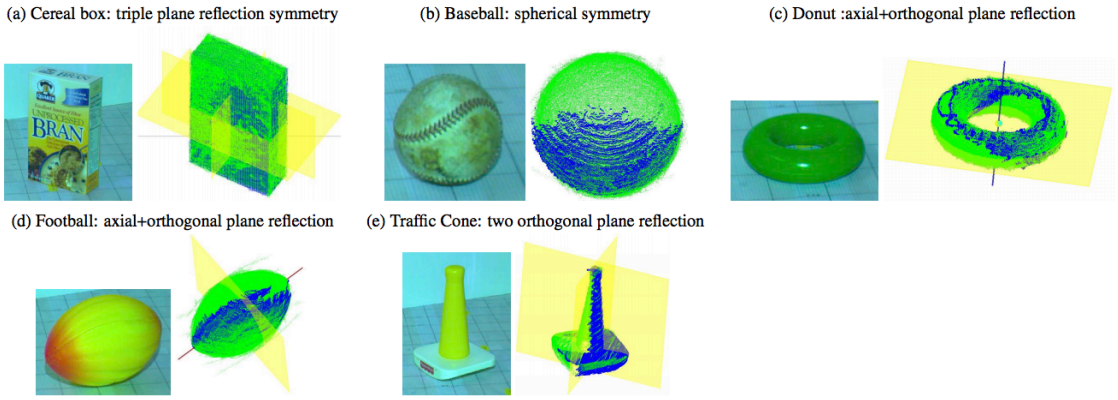


Figure 2-2: Shape completion from different types of symmetry [2]. Blue: partial observation; Green: shape from symmetry.

Thrun and Wegbreit [2] proposed shape completion from different types of symmetry, such as plane reflection, line reflection, point reflection, axial symmetry, spherical symmetry, and composite symmetry, etc. Given the partial data and the parameters of the symmetry plane, axis, or points, they applied the completion based on symmetry and used a probabilistic model to measure the likelihood of the partial observation being sampled from the complete model. The algorithm discretely searches the parameters for symmetry with hierarchy and optimize locally to fit the data better. When the objects are complex with multiple symmetric parts, the search space becomes prohibitively large. The algorithm needs more assumptions and restrictions to find the part symmetries.

Speciale *et al.* [15] detected symmetry plane by extracting features from the truncated signed distance function (TSDF) and matching the high-curvature and high-gradient regions with RANSAC or Hough transformation. After symmetry detection, instead of directly reflecting all the points, the completion is formulated as an optimization problem to minimize the cost with respect to surface smoothness and symmetry.

Sung *et al.* [16] further combined symmetry-based completion with database knowledge. They annotated the structure and symmetry in the database so that the incomplete scan can find the closest structure and perform symmetry-based completion. They also provide quantitative results by randomly removing points from complete point clouds and measure the Hausdorff distance.

The symmetry-based method has very impressive results. The limitation is that the huge searching space for symmetry detection makes the model require many assumptions to work efficiently. Combined with the database, the searching is accelerated, but it required a similar annotated structure existing in the database.

2.1.3 Gaussian Process Implicit Surface

In the robotics community, Gaussian Process is widely used because it could provide uncertainty for its prediction. To represent the 3D objects from sparse points, Gaussian Process Implicit Surface (GPIS) was proposed [3]. The GPIS combined Gaussian Process Regression with implicit surface, so that the 3D surface is represented with all the points whose Gaussian Process prediction is equal to 0. The GPIS also has the uncertainty for each prediction on the surface so that robots can actively explore the uncertain area.

Yi *et al.* [17] explored Gaussian Process Regression for 2.5D depth reconstruction from sparse data points. The robot explored the uncertain regions to work more efficiently. Kaul *et al.* [18] extended GPIS with surface normal constraint for shape completion. Bjorkman *et al.* [8] combined the visual and tactile data using GPIS to actively refine the 3D shape. Mahler *et al.* [19] explored how GPIS can be used with sequential convex programming for grasp planning.

GPIS can generate smooth surfaces and provide uncertainty for better active perception. The limitation is that to find the implicit surface, it's necessary to predict

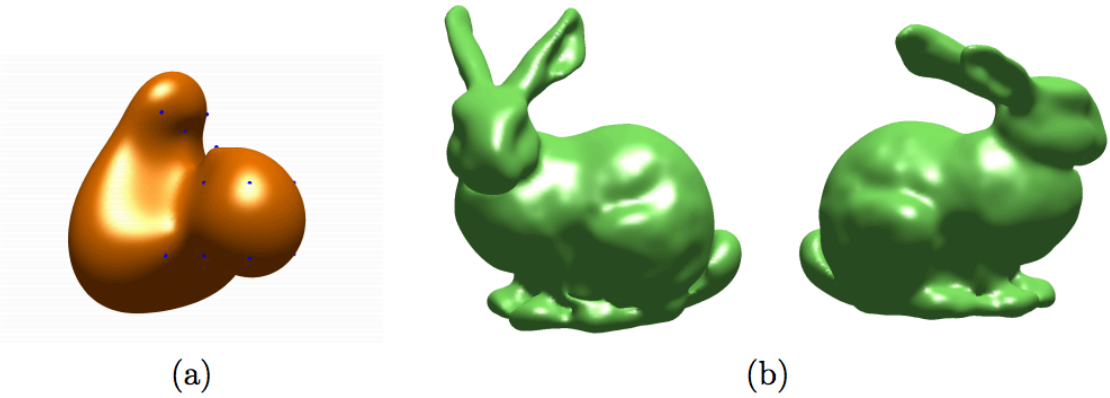


Figure 2-3: Gaussian Process Implicit Surface representing (a) A simple "blob" defined by 15 points on the surface, one interior +1 point and 8 exterior -1 points arranged as a cube; (b) Two views of the Stanford bunny defined by 800 surface points, one interior +1 point, and a sphere of 80 exterior -1 points [3]

most of the points in space which could be expensive if there are many data points collected [11].

2.1.4 Neural Network-based

With the recent advancements in deep learning and large 3D shape dataset [20], researchers explored to leverage the power of neural networks for shape completion. Wu *et al.* [21] was among the first to proposed 3D-CNN for shape recognition and completion. They represented the 3D shape in voxels which is convenient for neural networks to train and infer. Dai *et al.* [4] obtained very impressive results on 3D shape completion from partial depth scans by levering 3D convolutional networks and nonparametric patch-based shape synthesis methods. More recently, Varley *et al.* [10] explored how to better grasp an object by first employing a convolutional neural net for shape completion.

A more challenging problem is to recover 3D object shape from a single RGB image, without depth information. Solving the problem requires both powerful recognition systems and prior shape knowledge. With large-scale shape repositories like ShapeNet [20], researchers have made significant progress on data-driven approaches for shape synthesis, completion, and reconstruction [22, 23, 24, 25, 26].

The problem of 3D reconstruction from RGB data can be reduced into 3D shape completion by first estimating intrinsic images (*e.g.*, depth, and surface normal maps)

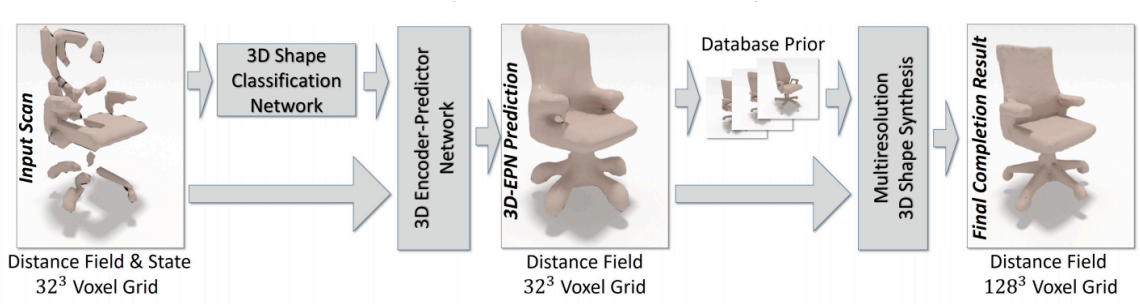


Figure 2-4: 3D shape completion using Convolutional Neural Networks (CNN) and 3D shape synthesis for refinement [4]

from RGB data [6]. Some recent papers have studied the problem of depth and surface normal estimation [27] from a single image.

In particular, the visual component of our model builds upon MarrNet [28], which jointly estimates intrinsic images and full 3D shape from a color image and has demonstrated good performance on standard benchmarks [29].

In this thesis, we propose to use intrinsic images as a unified representation for both visual and tactile observations and the learned shape priors. We demonstrate that our system is able to recover better 3D shape from either RGB or depth observations, compared to the state-of-the-art.

2.2 Tactile Sensing for Shape Reconstruction

In robotics, multi-modal learning has been widely exploited for grasping [30], tracking [31], scene layout probing [32], and shape recognition with active exploration [33]. There has been also research on connecting multi-modality data, *e.g.*, localizing object contact via visual observation [34], using vision to learn better tactile representations [35], and learning the sharing features between vision and tactile [36].

For shape reconstruction in particular, tactile data have also been exploited for both local [37][38] and global shape completion [39, 40], sometimes in a bimanual setting [41]. In recent years, researchers started to use active learning for shape reconstruction from tactile sensing [42, 17, 43, 44]. Luo *et al.* recently wrote a comprehensive review article on tactile perception which includes object shape perception [45].

Tactile data have been used to complement visual observations for shape recon-

struction [8], shape reasoning [46], and grasping [47]. Planning has also found its use in shape estimation from visual and tactile data [48]. We refer readers to Bohg *et al.* [7] for a thorough review. These papers, however, directly augment visual observations with tactile signals without leveraging shape priors. In comparison, we use shape priors learned from large-scale shape repositories to efficiently integrate tactile and visual observations.

Figure 2-5: (a) A cookie is pressed against the skin of an elastomer block. (b) The skin is distorted, as shown in this view from beneath. (c) The cookie's shape can be measured using photometric stereo and rendered at a novel viewpoint. [5]

In this thesis, we obtain tactile observations with the GelSight sensor [9]. The GelSight sensor is engineered mainly to achieve high precision for the measurement of the contact surface geometry [5] and shear force [49]. The GelSight sensor consists of three components: (1) soft silicone gel (2) color LEDs illumination and (3) a webcam. The three-color LEDs illuminate the gel from different angles. Since each surface normal corresponds to a unique color, the color image captured by the camera can be directly used to reconstruct the depth map of the contact surface by looking up a pre-made color-surface normal table. GelSight is able to recover high-fidelity object shape. This makes it particularly useful in object shape reconstruction among tactile sensors. GelSight has also found its in wide applications including physical material modeling [50] [51], surface hardness estimation [52], slip detection [53], and robot grasping [54].

Chapter 3

Method

We reconstruct the 3D shapes of the objects from both vision and touch. The pipeline of the system is described in Figure 3-1: we first reconstruct a voxelized rough 3D model of the object from a Kinect color image, and then touch the areas that visual prediction is not of high confidence. The tactile data provide us with the precise location and geometry of the object surface, especially in the occluded areas. These signals can later be posed as constraints to refine the 3D shape. The touch is conducted in a closed-loop exploration process: each time the robot touches the surface location which has the maximum uncertainty in the shape prediction. The policy aims to reduce the times of touches, making the reconstruction more efficient.

3.1 3D Reconstruction from Vision and Shape Priors

Our 3D reconstruction model exploits a key intermediate representation—intrinsic images (*a.k.a.* 2.5D sketches) [6]. The use of intrinsic images brings in two key advantages. First, it is a unified representation that integrates multi-modal data (RGB images, depth maps, and tactile signals). Using intrinsic images allows us to build a principled framework for multi-model shape reconstruction. Second, color images and 3D shapes become conditionally independent given intrinsic images. When depth data are not available, our formulation decomposes the challenging problem of single-image 3D reconstruction into two simpler ones: intrinsic image estimation and 3D

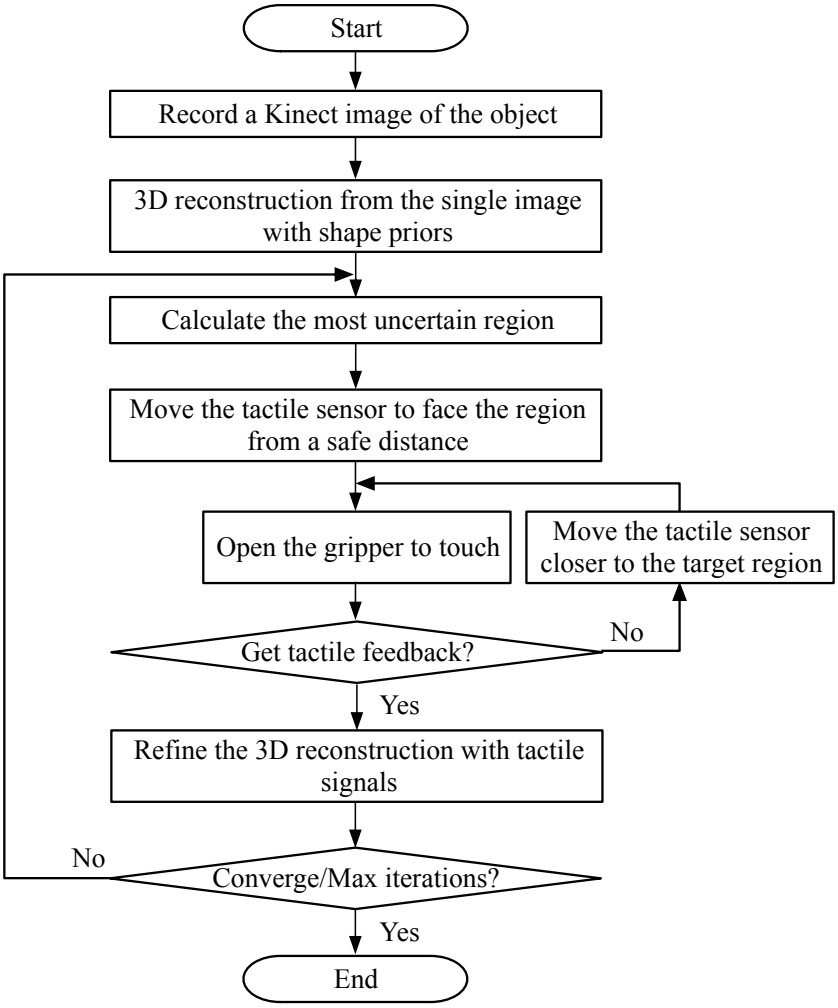


Figure 3-1: An overview of our interactive system that estimates 3D shape from monocular vision, touch, and shape priors.

shape completion. This provides us with better reconstruction results from a color image.

Our network, therefore, has two components to recover 3D shape from a color image. The first is a 2.5D sketch estimator (Figure 3-2-I), predicting the object's depth, surface normals, and silhouette from the color image; The second is a 3D shape estimator (Figure 3-2-II), inferring voxelized 3D object shape from intrinsic images. When depth data is available, we can use them to replace the predicted depth for possible better performance.

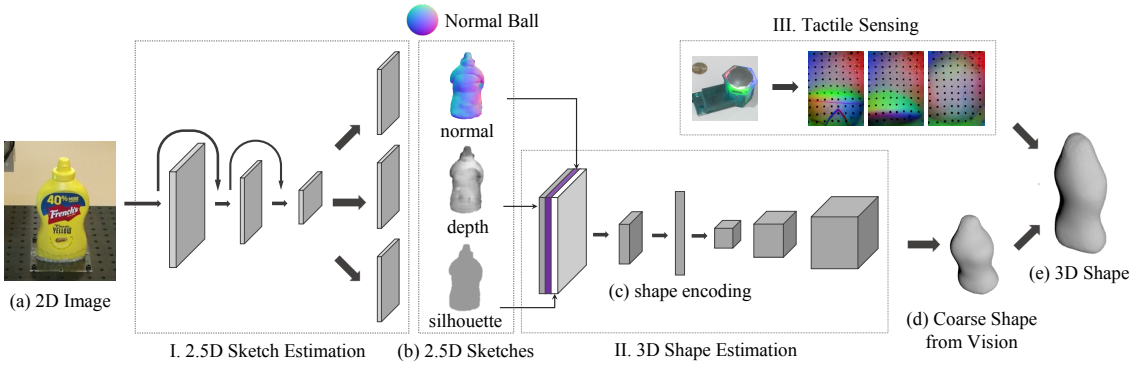


Figure 3-2: Our model has three major components. It first estimates the object's 2.5D sketches (depth, surface normals, and silhouette) from a single RGB image. It then recovers a rough 3D shape from them. Third, it integrates tactile signals to update the latent shape encoding and to generate a refined 3D shape.

3.1.1 2.5D Sketch Estimation

The first component of our network (Figure 3-2-I) takes a 2D color image as input and predicts its 2.5D sketches: depth, surface normals, and silhouette. The goal of 2.5D sketch estimation is to distill intrinsic object properties from input images, while discarding properties that are non-essential for the task of 3D reconstruction, such as object texture and lighting.

We use an encoder-decoder network for this step. Our encoder is a ResNet-18 [55], turning a 256×256 RGB image into 384 feature maps, each of size 16×16. Our decoder has three branches for depth, surface normals, and silhouette, respectively. Each branch has four sets of 5×5 transposed convolutional, batch normalization, and ReLU layers, followed by a 1×1 convolutional layer. It outputs at the resolution of 256×256.

3.1.2 3D Shape Estimation

The second module (Figure 3-2-II) infers 3D object shape from estimated 2.5D sketches. Here, the network focuses on learning priors of common shapes. The network architecture is again an encoder and a decoder. It takes a normal image and a depth image as input (both masked by the estimated silhouette), maps them to a 200-dim vector via a modified version of ResNet-18 [55]. We changed the average pooling layer into an adaptive average pooling layer, and the output dimension of the last

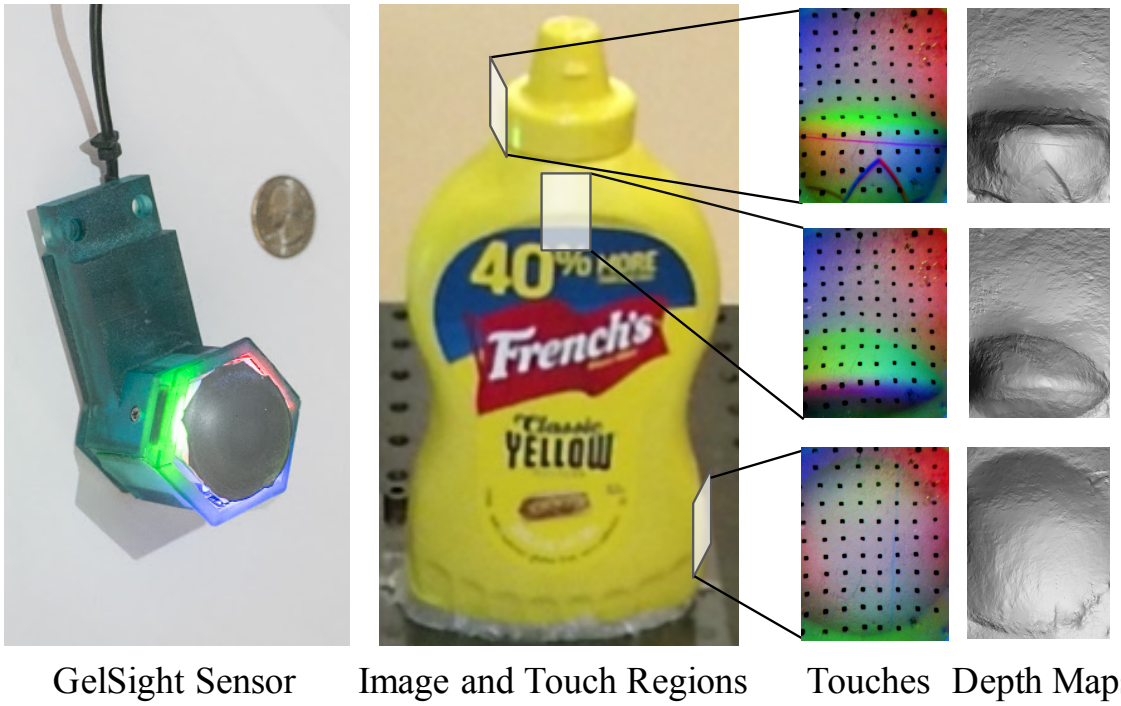


Figure 3-3: Tactile signals on different parts of the object and the corresponding 3D reconstructions

linear layer to 200. The vector then goes through a decoder, consisting of five sets of transposed convolutional, batch normalization, and ReLU layers followed by a transposed convolutional layer and a sigmoid layer to output a 128×128×128 voxel-based reconstruction of the shape.

3.2 Tactile Sensing for Shape Refinement

Tactile sensing obtains precise information in the local area: the data from the GelSight sensor provide high-resolution 3D geometry of the contact surface, and the position reading from the robot tells the exact location of the touch surface in the global space. The tactile data set solid constraints on the object's shape, and thus help to refine the 3D shape prediction from vision.

3.2.1 3D Reconstruction from GelSight

We can reconstruct the height function $z = f(x, y)$ from the GelSight tactile image [9]. Under the assumption that the lighting and surface reflectance are evenly distributed, the light intensity $\boldsymbol{I}$ at (x, y) can be modeled as

$$\mathbf{I}(x, y) = \mathbf{R}\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{3.1}$$

where $\boldsymbol{R}$ is the reflectance function which is a nonlinear function.

We first build a lookup table to obtain the inverse function $\boldsymbol{R}^{-1}$, which maps observed intensity to geometry gradients. A ball with known radius is pressed on the GelSight multiple times to collect data. Then, the gradient can be computed as

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = \boldsymbol{R}^{-1}(\boldsymbol{I}(x, y)) \tag{3.2}$$

After calculating the gradients, we reconstruct the height map $z = f(x, y)$ by integrating the gradients. It can be represented as the Poisson equations $(\nabla f)^2 = g$, where

$$g = \frac{\partial f}{\partial x}\left(\frac{\partial f}{\partial x}\right) + \frac{\partial f}{\partial y}\left(\frac{\partial f}{\partial y}\right). \tag{3.3}$$

We use the fast Poisson solver with the discrete sine transform (DST) to solve it, and get the height-map reconstruction. Figure 3-3 shows some examples of the GelSight images and the reconstructed 3D surfaces when contacting different areas on the mustard bottle.

3.2.2 Registration of World and System Coordinates

We need to register three coordinate systems: world, robot, and voxel (vision). To align the world the robot frame, we calibrate three points in the real world $\mathbf{x_{w1}} = (0, 0, 0)^T$, $\mathbf{x_{w2}} = (1, 0, 0)^T$, $\mathbf{x_{w3}} = (0, 1, 0)^T$, record their corresponding robot coordinates $\mathbf{x_{r1}}$, $\mathbf{x_{r2}}$, $\mathbf{x_{r3}}$, and calculate the transformation matrix by solving the linear equations

$$\mathbf{X_r} = \mathbf{R_r} \cdot \mathbf{X_w} + \mathbf{T_r}, \tag{3.4}$$

where $\mathbf{X_r} = [\mathbf{x_{r1}}, \mathbf{x_{r2}}, \mathbf{x_{r3}}]$, $\mathbf{X_w} = [\mathbf{x_{w1}}, \mathbf{x_{w2}}, \mathbf{x_{w3}}]$, $\mathbf{R_r}$ is the rotation matrix, and $\mathbf{T_r}$ is the translation vector.

To align voxels with the world frame, we use the correspondence of a fixed point $\mathbf{o}$, axes $\mathbf{a_x}, \mathbf{a_y}, \mathbf{a_z}$, and the scale s to calculate the transformation. The bottom center of the voxels $\mathbf{o_v}$ is aligned with the fixed point on the table in the world frame $\mathbf{o_w}$. The axes can be calculated based on the camera's position and orientation. In our setting, $\mathbf{a_x} = (-1, 0, 0)^T$, $\mathbf{a_y} = (0, 1, 0)^T$, $\mathbf{a_z} = (0, 0, 1)^T$. The scale of each voxel can be calculated by $s = n_p \times l_p$, where n_p is the number of corresponding pixels to each voxel and l_p is the length of each pixel in the real world. Then the rotation matrix $\mathbf{R_v} = s \cdot \left[\mathbf{a_x}, \mathbf{a_y}, \mathbf{a_z}\right]^T$. The transformation between world and voxel coordinate can be represented as

$$\mathbf{x_w} = \mathbf{R_v} \cdot (\mathbf{x_v} - \mathbf{o_v}) + \mathbf{o_w}. \tag{3.5}$$

After registration, we can map touches into corresponding voxels and control the robot arm to touch the target regions in the real world.

3.2.3 Updating Shape Reconstruction with Touch

We then present how we update the model's prediction with tactile signals, after converting them into surface normals, and registering them into the system coordinates. The key observation here is to design a differentiable loss function that enables fine-tuning with back-propagation.

Figure 3-4 illustrates our design. Given a 3D point in space and its normal vector $\boldsymbol{n}$, we gradually move the robot arm toward the destination, unless it touches a solid object halfway between. Either way, we obtain signals on whether the 3D voxels along the trajectory are occupied. We use $v_{\boldsymbol{p}}$ to represent the value at position $\boldsymbol{p}$ in a 3D voxel grid, where $v_{\boldsymbol{p}} \in [0, 1]$. Assume the GelSight sensor suggests the voxel $\boldsymbol{p}_0 = \{x_0, y_0, z_0\}$ is filled (Figure 3-4b). Our differentiable loss tries to encourage the voxel's value to be 1, and all voxels in front of it, along the direction $\boldsymbol{n}$, to be 0. This ensures the estimated 3D shape matches the obtained tactile signals. The

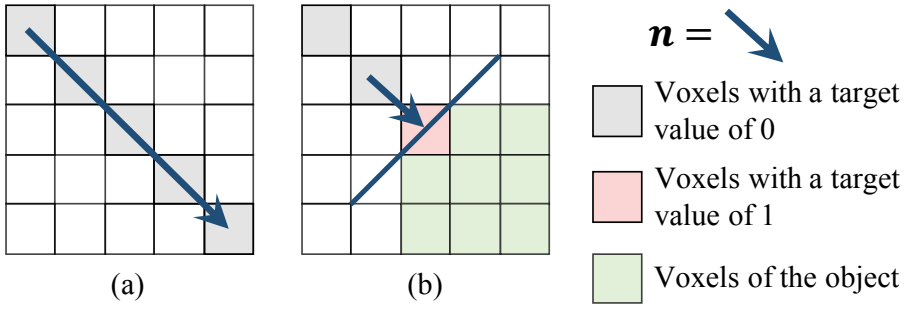


Figure 3-4: Reprojection loss for touches. (a) When the sensor makes a touch attempt but fails to reach the object, the voxels along its trajectory should all be 0. (b) When the sensor contacts the object, the corresponding voxels should be 1, and all voxels in front of it along the trajectory should be 0.

differentiable loss for a voxel $\boldsymbol{p}$ is defined as

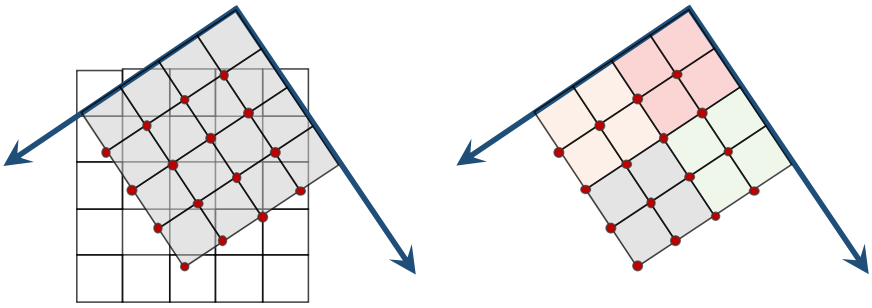
$$L(v_{\boldsymbol{p}}) = \begin{cases} v_{\boldsymbol{p}}^2, & \boldsymbol{p} = \boldsymbol{p}_0 + k\boldsymbol{n}, \quad \forall k < 0 \\ (1 - v_{\boldsymbol{p}})^2, & \boldsymbol{p} = \boldsymbol{p}_0 \\ 0, & \text{otherwise} \end{cases}. \tag{3.6}$$

The gradients are

$$\frac{\partial L(v_{\boldsymbol{p}})}{\partial v_{\boldsymbol{p}}} = \begin{cases} 2v_{\boldsymbol{p}}, & \boldsymbol{p} = \boldsymbol{p}_0 + k\boldsymbol{n}, \quad \forall k < 0 \\ 2(v_{\boldsymbol{p}} - 1), & \boldsymbol{p} = \boldsymbol{p}_0 \\ 0, & \text{otherwise} \end{cases}. \tag{3.7}$$

The loss and gradients can be similarly derived when the GelSight sensor suggests the voxel $\boldsymbol{p}_0$ is empty (Figure 3-4a).

After collecting touch signals, we compute losses and back-propagate gradients to the latent vector from the 2.5D sketch encoder. We then update it (with a learning rate of 0.001) and use the shape decoder to get a new shape. We repeat this process for 10 iterations for each touch.



(a) Overlaid search grid (b) Integral maps for region search

Figure 3-5: Our policy on finding the next place to touch. (a) A 2D search grid overlaid on the voxel grid, where the confidence values of the voxel prediction are assigned to the search grid. (b) After the assignment, we compute the integral map and use it to efficiently search for the region of maximal uncertainty. See text for details.

3.3 Policy for Active Tactile Exploration

We here describe our policy that automatically discovers the most uncertain region of the prediction for the next tactile exploration. Since the value of each voxel $v_{i,j,k}$ is the output of the sigmoid function which indicates the existing probability, the network's confidence score of voxel $v_{i,j,k}$ is defined as $c_{i,j,k} = |v_{i,j,k} - 0.5|$. We therefore would like to find a region S that is of the same size as the GelSight sensor and minimizes $\sum_{(i,j,k)\in S} c_{i,j,k}$.

This seemingly simple problem is challenging as the region S can be of any orientation, and we want the optimization to be fast. Our algorithm is based on integral maps. Given a plane, we sample a 2D grid on the plane and assign each point's confidence score $f_{p,q}$ as its closest voxel's confidence score, as shown in Figure 3-5a. We then compute the integral maps on the 2D grid; specifically, we have $g_{p,q} = \sum_{i=1}^{p} \sum_{j=1}^{q} f_{i,j}$. As

$$g_{p,q} = f_{p,q} + g_{p-1,q} + g_{p,q-1} - g_{p-1,q-1}, \tag{3.8}$$

we can compute the matrix $\boldsymbol{G}$ in $O(N^2)$ time, where N is the length of the voxel grid.

As the size of the GelSight sensor $S = k \times k$ is known, we can then find the region S with a minimal summed confidence score using $\boldsymbol{G}$, again in $O(N^2)$. This is because for a particular region $[p+1, p+k] \times [q+1, q+k]$, as shown in Figure 3-5b, we can

compute its regional sum in $O(1)$ as

$$\sum_{i=1}^{k}\sum_{j=1}^{k} f_{p+i,q+j} = g_{p+k,q+k} - g_{p,q+k} - g_{p+k,q} + g_{p,q}. \tag{3.9}$$

Finally, we in parallel evaluate multiple planes by searching over yaws (every 90°) and pitches (every 10°).

Chapter 4

Experiments

We now present experimental results. We first introduce our robot platform setup and how we generate training data for the networks. We then discuss our main results—how we reconstruct high-quality 3D shapes with vision, touch, and shape priors. Further, we conduct ablation studies to understand the contributions of each model component: how shape priors and the active exploration policy help to reconstruct shapes more efficiently, and how well our system adapts to RGB and depth data.

4.1 Robotic System Setup

The robotic system includes a 6-DOF robot arm, a GelSight tactile sensor, and a Kinect 2 (as shown in Figure 1-1). The GelSight sensor is mounted on a WSG 50 parallel gripper for the convenience. The target object is fixed to an optical breadboard in the robot's working space so that it will keep static during the interaction with the robot.

The robot arm is a UR5 from Universal Robotics with a reach radius of 850mm. The WSG 50 gripper is a parallel gripper from Weiss Robotics with force feedback. We do not use the gripper for gripping the objects, but we use the gripper's force feedback to alert collision of the sensor so that we install the GelSight sensor outwards in order to better touch the objects. The GelSight sensor we apply is the version introduced in [56]. It captures the surface geometry of a contact area of 19mm×14mm with a

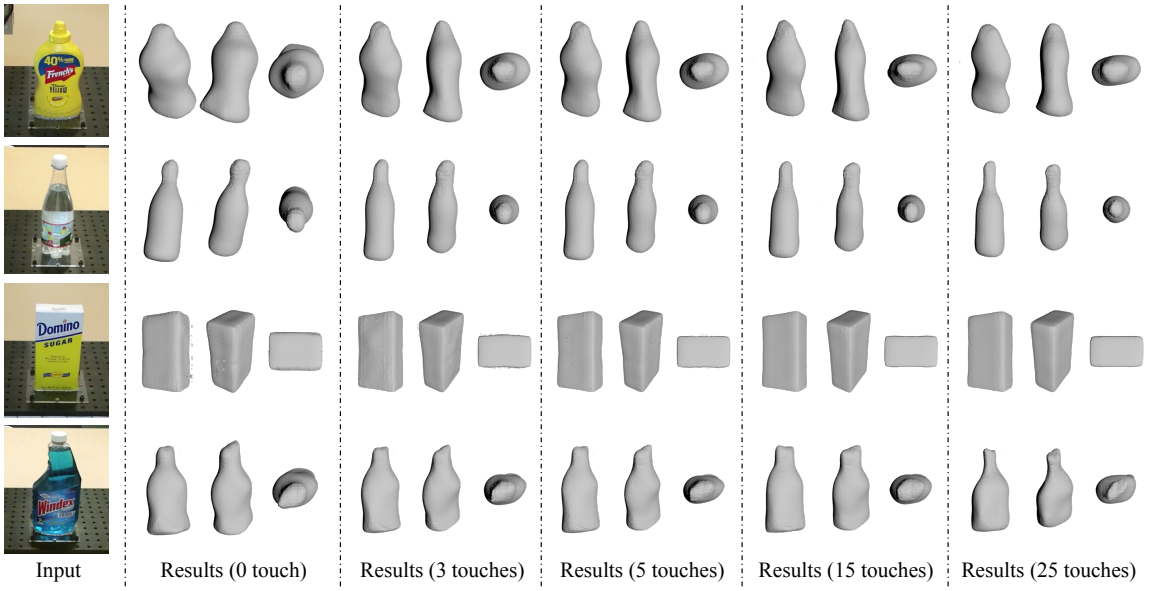


Figure 4-1: Results on 3D shape perception. From a single RGB image, our model recovers a rough 3D shape using shape priors. The reconstruction often captures the basic geometry, but deviates from the actual shape in various ways. The results improve gradually with touch signals. For example, for the bell-shaped bottle in the last row, the initial reconstruction is too fat (best seen from the top-down view). With tactile signals, our model recovers its flat shape. Our system also corrects object pose, as shown in the water bottle case.

resolution of 640×480 and a frequency of 30Hz. The raw output from the sensor is in the format of an image, and we reconstruct the 2.5D topography of the surface from it. The Kinect 2 captures RGB images of the target area, and is fixed on the side of the table at a 45.72cm height and a 30° tilt angle.

4.2 Conducting Touch without Collision

When touching the object surface, the robot should carefully avoid collision with the object. This is especially the case in our setup, as the initial 3D reconstruction can be imprecise, and the robot does not have much effective contact feedback other than the sensing surface of the GelSight sensor. We make the robot progressively head toward the target region from distance in the direction of the surface normal. In each touch attempt, the approach is conducted by the slow opening of the parallel gripper, so that the force feedback from the gripper's current provides a protection of the collision, especially when the collision does not happen on the GelSight's sensing area. At the same time, we also plan the motion of the robot when transferring between

different touch attempt to avoid interfering with the object. Our basic strategy is to take a detour in the high-up area when changing the target positions. But we also calculate the radial angles between the two target locations. When the angle is small, it indicates that the two locations are close, and it is safe for the robot to move directly to the second location to save time.

4.3 Dataset

We generate synthetic training data of paired images and 3D shapes for networks to learn shape priors. We use Mitsuba [57] to render fourteen object categories (bag, bottle, bowl, camera, can, cap, computer keyboard, earphone, helmet, jar, knife, laptop, mug, remote control) in ShapeNet [20] from 20 random views using three types of backgrounds: 1/3 on a clean, white background, 1/3 on high-dynamic-range backgrounds with illumination channels, and 1/3 on backgrounds randomly sampled from the SUN database [58]. For each object in each view, we render an RGB image and its depth, surface normal, and silhouette. We augment our training data by color and light jittering during training.

We train the 2.5D sketch estimator and the 3D shape estimator separately on synthetic images. The 2.5D sketch estimator is trained using the ground truth surface normal, depth, and silhouette images with an L2 loss. The 3D shape estimator is trained using ground truth voxels and a binary cross-entropy loss. We implement our model in PyTorch. We use the Adam optimizer [59] with $\beta_1 = 0.5$, $\beta_2 = 0.9$ and a learning rate of 5×10^{-4} for the 2.5D sketch estimator, and stochastic gradient descent with a learning rate of 2×10^{-2} and a momentum of 0.9 for the 3D shape estimator. For visualization, bilateral filters are applied to remove aliasing [60].

4.4 Results

We show the main results in Figure 4-1. From a single RGB image, our learned model correctly segments the object and produces a rough 3D shape estimation. We then let the robot automatically touch the objects and use the tactile signals to further refine the shape. For the sugar box in row 3, we use a prior learned on box-like shapes

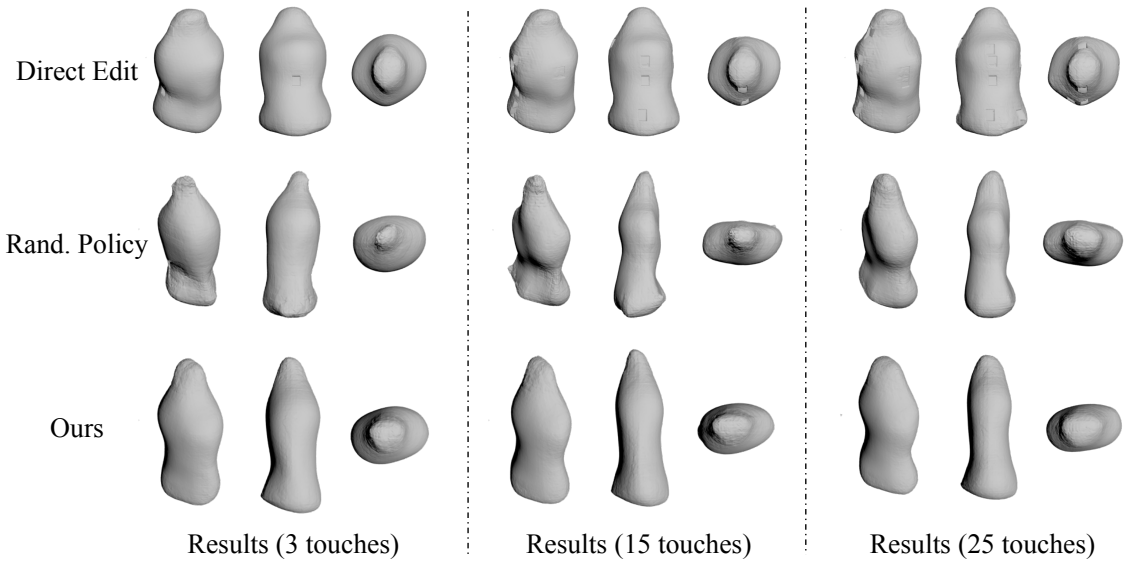


Figure 4-2: We show the effects of shape priors and the policy. If we Direct Edit the voxels' value (not using learned priors to update), each touch can only be used to update the shape locally. The shape does not change much even after many touches. With Random Policy, it takes longer for the model to obtain fine shape structure.

instead of all fourteen categories. An ablation study is presented in Section 4.5.

Our system works well on a variety of object shapes. Each example shown in the figure has its distinct shape, and our model works well on all of them. For example, our model recovers the fine curvature of the spray bottles. As our model does not require a depth image as input, it can deal with transparent objects like the water bottle (though it can still use Kinect depth when available, as shown in Section 4.6).

4.5 Shape Priors and the Exploration Policy

We then present three ablation studies to understand how the learned priors and the active exploration policy contribute to its final performance. First, we compare our model with two variants: Direct Edit and Random Policy. The first one does not use shape priors; instead, it directly uses the tactile signals to edit the voxelized shape, *i.e.* changing the values of the touched voxels to 1 and the voxels in front of them to 0. The second does not use our policy. It randomly chooses where to touch within the object's bounding box. The performance of the second baseline has large variance due to its randomness. For quantitative evaluation, we run it 10 times and compute

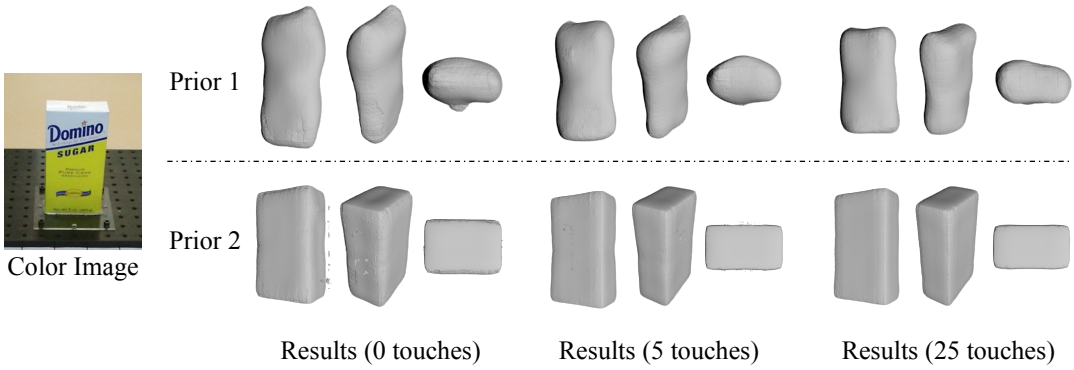


Figure 4-3: The two priors on the sugar box. A network trained on general shapes predicts a less accurate shape, which is later corrected by touches. A network trained on box-like shapes gives better results.

the mean of its scores.

Figure 4-2 shows qualitative results. Both the policy and the shape priors help to obtain an accurate shape estimation much faster, significantly reducing the number of touches required. Without the priors, each touch can only be used to update a local region of the shape; without the policy, the shape may become significantly worse before eventually getting better.

We further quantitatively compare the shape obtained after each update with the ground truth shape. Our metric is the classic Chamfer distance (CD) [61], widely used in the computer graphics community for measuring shape similarity. For each point in each cloud, CD finds the nearest point in the other point set, and sums the distances up.

We show quantitative results in Figure 4-4. Here, we also have a human policy, where humans select the position of the next touch. This can be seen as a reference for the optimal policy. Our full model achieves a low Chamfer distance after a few touches, close to the human, while the baselines (w/o policy or priors) take much longer.

We also evaluate how priors learned on different training sets affect results. Figure 4-3 shows that for the sugar box, a network trained on general shapes predicts a less accurate shape, which is later corrected by touches; in contrast, a network trained on box-like shapes gives better results. This reveals an interesting future direction: it will be helpful to classify the object's type from vision, which may inform the most

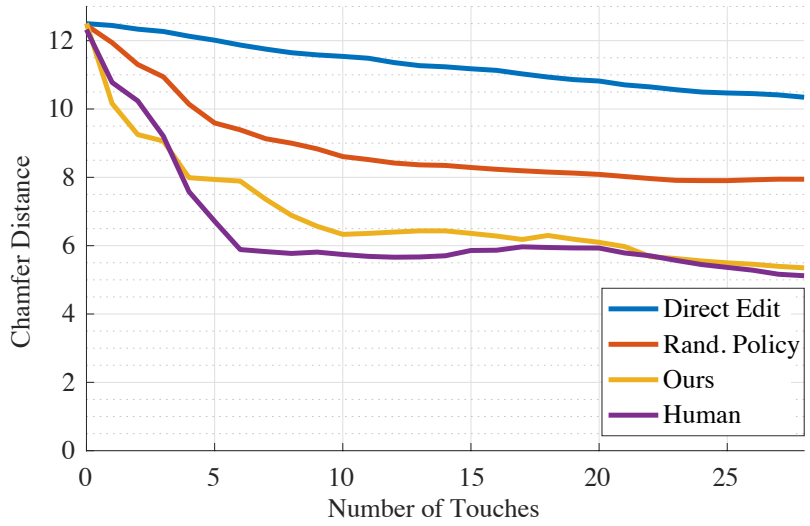


Figure 4-4: Shape estimation accuracy with respect to the number of touches, measured in Chamfer distance. Our policy recovers the shape accurately and efficiently. With Random Policy, it takes much longer to reconstruct a reasonable shape; if we Direct Edit the voxels' value (not using learned priors to update), the object is hardly updated after each touch. The Human method asks a human to manually select where to touch for each step and can be seen as a reference for comparison.

efficient policy and prior.

4.6 RGB vs RGB-D Input

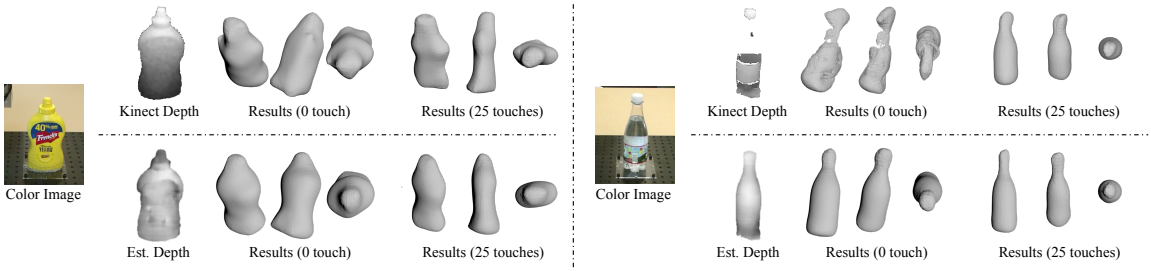


Figure 4-5: Our method can use either our estimated depth maps or Kinect depth maps. A Kinect depth map can be helpful if it is accurate: for example, the initial reconstruction of the left bottle is flatter using the Kinect depth map. However, if we purely rely on Kinect depth, our reconstruction would not be as accurate when the Kinect depth is inaccurate (see the transparent water bottle).

We finally evaluate how our model works on RGB *vs.* RGB-D data, to better understand its practical applicability. Figure 4-5 reveals that our method can use either our estimated depth maps or Kinect depth maps. A Kinect depth map can

be helpful if it is accurate: for example, the initial reconstruction of the left bottle is flatter (and therefore better) using the Kinect depth map. However, Kinect depth maps can also be unreliable: it fails to estimate the depth of the transparent water bottle. If we purely rely on Kinect depth, our reconstruction would not be as accurate as our current formulation, which is able to recover 3D shape purely from a color image and touch.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We have presented a novel model for 3D shape perception that integrates visual and tactile signals with learned shape priors. Our model uses intrinsic images as the intermediate representation to unify multi-modal signals. We have also proposed an active exploration policy to search for the most informative touches. Our model performs well on real objects, recovering their 3D shape accurately. Ablation studies verify that the use of touch priors and the exploration policy enables more efficient shape recovery. Our model works well with RGB and RGB-D data, and can handle transparent objects.

We hope our approach can inspire future research in fusing common sense knowledge into building object models: the idea of learning an object prior can be extended to not only model shapes, but objects' physical attributes; we can also refine the learned object prior through interaction [7].

5.2 Future Work

There are many aspects of this work could be improved in the future, including how to better evaluate efficiency of the system, how to solve the immobilization problem and how to evaluate the reconstructed shape on real robot tasks.

5.2.1 Efficiency

Evaluate efficiency by real time instead of number of touches.

In this work, to evaluate the efficiency of the policy, we followed some previous works using accuracy with respect to the number of touches. However, in real experiments, the policy tends to explore regions back and forth, which takes a lot of time of switching between the regions repeatedly. A more interesting problem would be how long does the data collection really takes.

By considering the switching time, the policy would find the trade-off between uncertainty and distance from current position. Exploring a nearby regions take much less time than exploring a distant region then coming back in the middle.

Also sliding would be much more efficient than poking discretely. The question is how to find a safe exploration strategy for sliding. Torque information may be needed to avoid collistion. And increasing the tactile area of a hand can also make data collection more efficient.

5.2.2 Immobilization

Most of the works on 3D shape reconstruction from tactile would assume the object is fixed. To make the method more practical, this immobilization has to be solved.

One solution is using bimanual robots, so that another hand could grasp the object. Sommer *et al.* [41] explore the 3D reconstruction in bimannual setting. It would be interesting whether adding the high-resolution tactile sensors like GelSight could further improve the performance.

Another direction would be exploring the object dynamically, with tracking and matching the models during intereaction. Ilonen *et al.* [62] explored the direction of 3D reconstruction while grasping.

5.2.3 Evaluation on Robot Tasks

Evaluate how the reconstructed shape would be helpful for improving grasp planning and manipulation task.

In terms of what is a good estimated 3D shape, we evaluated the distance between the reconstruction and the groud-truth shape. Moreover, to actually help robotics

task, whether the fine details and smoothness of the object are important for robot could be further explored.

Bibliography

[1] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012.

[2] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[3] Oliver Williams and Andrew Fitzgibbon. Gaussian process implicit surfaces. *Gaussian Proc. in Practice*, pages 1–4, 2007.

[4] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Micah K Johnson and Edward H Adelson. Retrographic sensing for the measurement of surface texture and shape. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1070–1077. IEEE, 2009.

[6] Harry G Barrow and Jay M Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978.

[7] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Trans. Robotics*, 33(6):1273–1291, 2017.

[8] Marten Bjorkman, Yasemin Bekiroglu, Virgile Hogman, and Danica Kragic. Enhancing visual perception of shape through tactile glances. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

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

[10] Jacob Varley, Chad DeChant, Adam Richardson, Avinash Nair, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[11] David Watkins-Valls, Jacob Varley, and Peter Allen. Multi-modal geometric learning for grasping and manipulation. *arXiv preprint arXiv:1803.07671*, 2018.

[12] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012.

[13] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. *CGF*, 34(2):435–446, 2015.

[14] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.*, 25(3):560–568, 2006.

[15] Pablo Speciale, Martin R Oswald, Andrea Cohen, and Marc Pollefeys. A symmetry prior for convex variational 3d reconstruction. In *European Conference on Computer Vision*, pages 313–328. Springer, 2016.

[16] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Trans. Graph.*, 34(6):175, 2015.

[17] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. Active tactile object exploration with gaussian processes. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[18] Lukas Kaul, Simon Ottenhaus, Pascal Weiner, and Tamim Asfour. The sense of surface orientation—a new sensor modality for humanoid robots. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 820–825. IEEE, 2016.

[19] Jeffrey Mahler, Sachin Patil, Ben Kehoe, Jur Van Den Berg, Matei Ciocarlie, Pieter Abbeel, and Ken Goldberg. Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[20] Angel X Chang et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

[21] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Neural Information Processing Systems (NIPS)*, 2016.

[24] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning shape priors for 3d shape completion and reconstruction. In *European Conference on Computer Vision (ECCV)*, 2018.

[25] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D Kulkarni, and Joshua B Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] Michael Janner, Jiajun Wu, Tejas D. Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. In *Neural Information Processing Systems (NIPS)*, 2017.

[28] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Neural Information Processing Systems (NIPS)*, 2017.

[29] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] Peter K. Allen, Andrew T. Miller, Paul Y. Oh, and Brian S. Leibowitz. Integration of vision, force and tactile sensing for grasping. *Int. J. Intelligent Machines*, 4:129–149, 1999.

[31] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[32] Jeannette Bohg, Matthew Johnson-Roberson, Mårten Björkman, and Danica Kragic. Strategies for multi-modal scene exploration. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[33] Pietro Falco, Shuang Lu, Andrea Cirillo, Ciro Natale, Salvatore Pirozzi, and Dongheui Lee. Cross-modal visuo-tactile object recognition using robotic active exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[34] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. Localizing the object contact through matching tactile features with visual map. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[35] Oliver Kroemer, Christoph H Lampert, and Jan Peters. Learning dynamic tactile sensing with robust vision-based training. *IEEE Trans. Robotics*, 27(3):545–557, 2011.

[36] Shan Luo, Wenzhen Yuan, Edward Adelson, Anthony G Cohn, and Raul Fuentes. Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[37] Simon Ottenhaus, Martin Miller, David Schiebener, Nikolaus Vahrenkamp, and Tamim Asfour. Local implicit surface estimation for haptic exploration. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2016.

[38] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. Iterative closest labeled point for tactile object shape recognition. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[39] Alexander Bierbaum, Ilya Gubarev, and Rüdiger Dillmann. Robust shape recovery for sparse contact location and normal data from haptic exploration. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[40] Zachary Pezzementi, Caitlin Reyda, and Gregory D Hager. Object mapping, recognition, and localization from tactile geometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[41] Nicolas Sommer, Miao Li, and Aude Billard. Bimanual compliant tactile exploration for grasping unknown objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[42] Danny Driess, Peter Englert, and Marc Toussaint. Active learning with query paths for tactile object shape exploration. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[43] Nawid Jamali, Carlo Ciliberto, Lorenzo Rosasco, and Lorenzo Natale. Active perception: Building objects' models using tactile exploration. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2016.

[44] Uriel Martinez-Hernandez, Giorgio Metta, Tony J Dodd, Tony J Prescott, Lorenzo Natale, and Nathan F Lepora. Active contour following to explore object shape with robot touch. In *World Haptics Conference (WHC)*, 2013.

[45] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[46] Jacob Varley, David Watkins, and Peter Allen. Visual-tactile geometric reasoning. In *RSS Workshop*, 2017.

[47] Jarmo Ilonen, Jeannette Bohg, and Ville Kyrki. Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. *Int. J. Robotics Res.*, 33(2):321–341, 2014.

[48] Takamitsu Matsubara and Kotaro Shibata. Active tactile exploration with uncertainty and travel cost for fast shape estimation of unknown objects. *Robotics Auton. Syst.*, 91:314–326, 2017.

[49] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015.

[50] Wenzhen Yuan, Shaoxiong Wang, Siyuan Dong, and Edward Adelson. Connecting look and feel: Associating the visual and tactile properties of physical materials. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward Adelson. Active clothing material perception using tactile sensing and deep learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[52] Wenzhen Yuan, Chenzhuo Zhu, Andrew Owens, Mandayam A Srinivasan, and Edward H Adelson. Shape-independent hardness estimation using deep learning and a gelsight tactile sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[53] Jianhua Li, Siyuan Dong, and Edward Adelson. Slip detection with combined tactile and visual information. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7772–7777. IEEE, 2018.

[54] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In *Conference on Robot Learning (CoRL)*, 2017.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[56] Siyuan Dong, Wenzhen Yuan, and Edward Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[57] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[58] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[60] Thouis R Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 943–949. ACM, 2003.

[61] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1977.

[62] Jarmo Ilonen, Jeannette Bohg, and Ville Kyrki. Fusing visual and tactile sensing for 3-d object reconstruction while grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.