Robotic Grasping through Combined Image-Based Grasp Proposal and 3D Reconstruction

Tarik Tosun, Daniel Yang, Ben Eisner, Volkan Isler, and Daniel Lee

***Abstract*— We present a novel approach to robotic grasp planning using both a learned grasp proposal network and a learned 3D shape reconstruction network. Our system generates 6-DOF grasps from a single RGB-D image of the target object, which is provided as input to both networks. By using the geometric reconstruction to refine the the candidate grasp produced by the grasp proposal network, our system is able to accurately grasp both known and unknown objects, even when the grasp location on the object is not visible in the input image.**

This paper presents the network architectures, training procedures, and grasp refinement method that comprise our system. Hardware experiments demonstrate the efficacy of our system at grasping both known and unknown objects (91% success rate). We additionally perform ablation studies that show the benefits of combining a learned grasp proposal with geometric reconstruction for grasping, and also show that our system outperforms several baselines in a grasping task.

I. Introduction

Object manipulation in unstructured environments remains a challenging problem in robotics. Recently, data-driven methods that estimate grasps directly from sensor data (without explicit intermediary state) have resulted in major advances in grasping performance and generalization [1]–[4]. Many of these methods operate on RGB-D image data, and employ convolutional neural network architectures that have proven effective at addressing tasks in computer vision [5], [6]. These systems approach grasping from a perception-driven perspective and often use pixel-space representations of the task, for example, learning grasp affordance maps over pixels [2] or constraining grasps to the image-plane normal [1].

While pixel-space representations have clear *computational* advantages, there are clear *physical* advantages to generating full 6-DOF grasps when interacting with real objects. In addition, in some contexts it is advantageous to infer the presence of a grasp point that is not directly visible in the observed image, either because it represents a favorable affordance for grasping (e.g. the handle of a mug) or because of the constraints of a task (e.g. grasping a visible point might make it difficult to place the object in a desired configuration after grasping). Many existing image-based data-driven methods formulate grasps by selecting a visible pixel at which to grasp, limiting the grasp plan to a visible point on the object [1], [2].

We propose a novel approach to grasp planning using both a learned *grasp proposal* network and a learned *shape reconstruction* network. Introducing explicit geometric reconstruction in addition to image-based grasp proposal allows us to fine-tune the proposed grasp for increased accuracy. Both networks operate on RGB-D images and employ state-of-the-art convolutional architectures, leveraging their representational power and generalization properties. In contrast to pixel-space methods, our grasp proposal network outputs a full 6-DOF grasp pose, and is capable of proposing grasps on occluded parts of an object not visible in the input image. Similarly, our 3D reconstruction network can infer the shape of the unseen portions of the object. Our system refines the proposed grasp by projecting it onto the reconstructed surface, enabling the robot to precisely grasp both hidden and visible portions of objects. In our experiments, we demonstrate that this refinement step improves grasping performance.

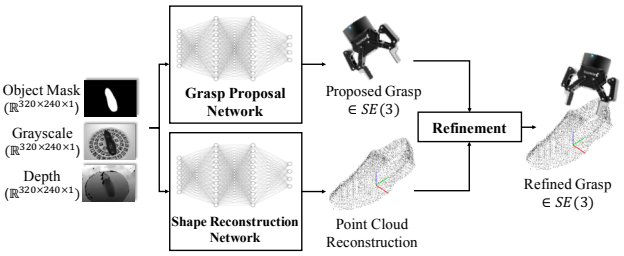

Fig. 1. System overview. An input RGB-D image with segmentation mask is provided as input to two neural networks that produce a 6-DOF grasp pose, and a 3D point cloud reconstruction of the object, respectively. The grasp pose is refined by projecting it onto nearest point in the point cloud, resulting in the final output grasp.

This paper presents the structure of our grasp planning system, including the details of the network architectures, training procedures, and data capture and labelling procedures for the grasp proposal and shape reconstruction networks. In hardware experiments, we demonstrate that our system is able to successfully grasp instances of an object category (shoes), including cases when the desired grasp point is not visible in the image, and also grasp object instances it was not trained on. Additionally, we include ablation studies to show the benefits of using both shape reconstruction and image-based grasp proposal, and also show that our system outperforms several baselines systems at the grasping task.

II. Related Work

Robotic grasping is a well-studied problem. Traditional methods for robotic grasp planning include analytic grasp planners that compute grasp stability metrics using physical models [7], as well as methods for online object pose estimation that employ optimization techniques to match

All authors are with the Samsung AI Center NY, 123 West 18th Street, New York, New York 10011

observed objects to a a library of known models [8].

Data-driven deep learning methods have shown promise in improving the efficacy and robustness of robotic grasping. Many of these methods operate on RGB-D data, and employ learning paradigms from computer vision. Some perception-driven methods learn visual features corresponding to affordances for grasping, represented as maps over image pixels [2], [9]. Others train neural networks that evaluate grasp quality as a function of gripper position and orientation given an input depth image [1], [10]. While these methods have demonstrated good performance at grasping known and unknown objects, the grasps they generate are constrained to lie on a visible surface of the object in the image. Our method generates full 6-DOF grasp poses, and can accurately grasp both hidden and visible portions of objects.

A recent method [3] generates grasp proposals by sampling using a variational autoencoder [11], and assesses and refines the grasp proposals using a second network that evaluates their quality. Our method operates under a similar principle of grasp proposal and refinement, but employs different techniques for both components. For grasp proposal, we perform a single inference step to regress a 6-DOF grasp from an input image using a convolutional neural network, where [3] samples hundreds of grasps using a VAE with a point cloud embedding network. For grasp refinement, we project the grasp proposal onto an explicit reconstruction of the target object, whereas [3] relies on a grasp quality network that implicitly models the interaction between the gripper and object.

Advances from the computer vision and graphics communities have produced methods for general shape reconstruction, which construct an estimated 3D representation of an object from an RGB or RGB-D image of the object [12]–[15]. Similar methods have been employed to enable robotic grasping [16], [17]. These methods use a neural network to reconstruct a 3D model of an object from a depth image, and then plan grasps using an analytic method. In our work, we employ a recent method that reconstructs an object as a point cloud from a partial-view RGB-D image [18].

Methods for category-level grasp planning include learned object representations using semantic 3D keypoints detected from RGB-D images [19], and representations that combine keypoints and shape completion [20]. The primary focus of these efforts is task representation, not grasping; grasps are formulated near keypoints, using a heuristic that takes into account the flatness of the local point cloud and antipodality of the grippers. In this paper, we study the effect of incorporating information 3D reconstruction on data-driven grasp planning.

III. Problem Statement

We consider a setup consisting of a robotic arm with known kinematics, an RGB-D camera, and an object to be grasped. We assume that a method of segmenting an image of the object from its background is available - this could be either a learned model for segmenting objects of a given category, or some other method (e.g. color based background subtraction, table-plane segmentation)

Throughout the paper, $\mathcal{R}$ denotes the base frame of the manipulator, $\mathcal{G}$ denotes the frame of the robot end-effector (a parallel-jaw gripper), $\mathcal{C}$ is the camera frame and $\mathcal{O}$ is the object body frame. The pose of the camera with respect to the gripper ${}^{\mathcal{G}}T_{\mathcal{C}}$ is assumed to be fixed and known, and the pose of the gripper with respect to the robot base ${}^{\mathcal{R}}T_{\mathcal{G}}$ is assumed to be computable via forward kinematics. Given a single, segmented RGB-D image of the object from the camera, the objective is for the robot to grasp and pick up the object. Picking is considered successful if at the end of the robot's motion, the object has been lifted completely off the table and is securely grasped by the gripper, and is neither damaged nor deformed by the grasp. In this paper, we train and benchmark our models using shoes as a representative class of objects.

IV. Methods

A. Overview

Our system (Figure 1) is comprised of two neural networks whose outputs are combined by a refinement module to plan grasps. The input to our system is a single RGB-D image which is captured, segmented, and provided to both networks.

The *grasp proposal network* `GPNet` outputs a *grasp pose* with respect to the camera frame, ${}^{\mathcal{C}}T_{\mathcal{G}} \in SE(3)$. The *shape reconstruction network* `SRNet` outputs a reconstructed point cloud of the object, providing a reasonable estimate of the shape of the occluded portions of the object. The outputs of the two networks are combined by projecting the grasp proposal ${}^{\mathcal{C}}T_{\mathcal{G}}$ onto the closest point in the reconstructed point cloud, resulting in a refined grasp proposal ${}^{\mathcal{C}}T_{\mathcal{G}}{}^{+}$. Since the pose of the camera with respect to the robot ${}^{\mathcal{R}}T_{\mathcal{C}}$ is known, this camera-frame grasp can be easily transformed into the robot frame for execution by the robot: ${}^{\mathcal{R}}T_{\mathcal{G}}{}^{+} = {}^{\mathcal{R}}T_{\mathcal{C}}{}^{\mathcal{C}}T_{\mathcal{G}}{}^{+}$.

B. Object Dataset

The dataset used to train `GPNet` is comprised of point cloud models, RGB-D images, masks, and poses of 34 real shoes. RGB-D images were gathered by placing shoes on a turntable and imaging them at 5-degree intervals using three Intel RealSense D435 cameras, for a total of 216 images per shoe. Images were segmented and fused to form dense, complete point clouds of each shoe, with known camera pose registration from each captured image to the complete point cloud. The dataset and data capture procedure are described in detail in [21].

C. Grasp Proposal Network - `GPNet`

1) Architecture: `GPNet` (Figure 2) consists of an embedding stage that processes a pair of aligned grayscale and depth images using parallel ResNet-34 convolutional modules [5], followed by two fully-connected layers that join the outputs of the RGB and Depth streams, similar in architecture to the network employed in [21]. The input images (I^g, I^d) are assumed to have been foreground masked,

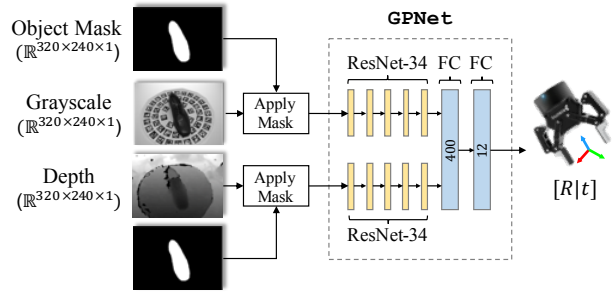

Fig. 2. The architecture of GPNet consists of parallel ResNet-34 modules that generate embeddings for a foreground-masked grayscale and depth image. These embeddings are concatenated, and regressed via two fully-connected layers into a vector, $\hat{t} \in \mathbb{R}^{12}$, representing a homogeneous transform, ${}^{\mathcal{C}}T_{\mathcal{G}}$.

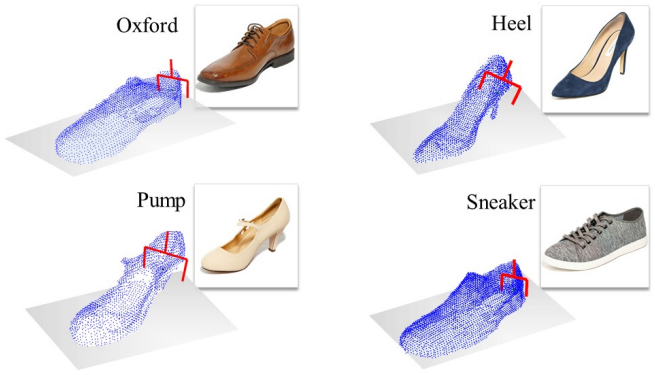

Fig. 3. Example grasps in our dataset. Using the 3D mesh of collected shoes and the CAD model of the gripper, GraspIt, a physics-based grasping simulator, generates a set of candidate grasps from which a single ground-truth example is selected for each shoe.

setting the non-shoe pixels and depth values to zero. The output of the network is a vector $\hat{t} \in \mathbb{R}^{12}$ representing a homogeneous transform ${}^{\mathcal{C}}\hat{T}_{\mathcal{G}}$, which is the estimated grasp pose (of the gripper with respect to the camera). The first 3 values (t_1, t_2, t_3) represent the desired (x, y, z) position of the gripper in the camera's coordinate frame. The last 9 values represent a serialized 3D rotation matrix giving the relative orientation of the gripper relative to the camera.

2) Example Grasps: Training GPNet requires a ground-truth pose of the gripper with respect to the camera (${}^{\mathcal{C}}T_{\mathcal{G}}{}^{*}$) for each RGB-D image in our dataset, but because the pose of the object with respect to the camera (${}^{\mathcal{C}}T_{\mathcal{O}}$) is recorded along with each dataset image, we only to generate a single object-frame example grasp ${}^{\mathcal{O}}T_{\mathcal{G}}{}^{*}$ for each *object* ($N = 25$ shoes) rather than for each *object image* ($N = 5394$). The example grasp can then simply be transformed into the camera frame: ${}^{\mathcal{C}}T_{\mathcal{G}}{}^{*} = {}^{\mathcal{C}}T_{\mathcal{O}}{}^{\mathcal{O}}T_{\mathcal{G}}{}^{*}$.

Ground-truth example grasps are generated for each shoe using GraspIt, a physics-based grasping simulator [7]. Each shoe point cloud is converted into a 3D mesh using the marching cubes algorithm [22], and provided as input to GraspIt, along with a CAD model of the gripper (Robotiq 2F-85). GraspIt samples, evaluates, and optimizes grasps via simulated annealing until the desired number of high-quality grasps have been generated (in our case, 100 grasps with a grasp contact energy below a specified threshold). From among these good grasps, a single ground-truth example ${}^{\mathcal{O}}T_{\mathcal{G}}{}^{*}$ is selected for each shoe. Selection can be done based on the desired behavior with respect to a task: for example, in the experiments presented in this paper, we manually select a grasp on the *left* side of the mouth of each example shoe, as shown in Figure 3. This might be appropriate for a scenario where the robot needs to grasp and arrange shoes on a shoe store display shelf, with the toe pointed to the left. If a semantically-consistent grasp is not required for a given application, this process could be automated by simply selecting the grasp with the highest quality score for each shoe.

3) Data Augmentation: Images in the dataset are collected from a turntable, so the shoe is always in the center of the image frame. To allow GPNet to generalize to images that are not dead-center in the image, we perform data augmentation at train time to randomly shift and scale the shoe in the to image, and apply corresponding geometric transformations the depth image to simulate the effect of the image-space scale/shift. As a result, we are able to expand our dataset of 5394 images to millions of randomly-augmented training examples. Full details of this augmentation technique are described in [21] .

4) Training: During training, the network is presented with input/output pairs $((I^g, I^d), t^*)$, where I^g and I^d are foreground masked grayscale and depth images, and t^* is the corresponding ground-truth grasp ${}^{\mathcal{C}}T_{\mathcal{G}}{}^{*}$, generated by the physics-based grasp planner and serialized as described in the previous section.

Given (I^g, I^d), the network makes a prediction $\hat{t}$ and a loss is computed from the ground truth grasp t^*. Let $\text{Trans}(t) : \mathbb{R}^{12} \rightarrow \mathbb{R}^3$ and $\text{Rot}(t) : \mathbb{R}^{12} \rightarrow \mathbb{R}^{3\times3}$ be functions that extract the position and rotation matrix components of 12-element vector t. The loss measures the closeness of the predicted and ground truth grasps, and is the weighted sum of a translation and a rotation component:

$$\ell(\hat{t}, t^*) = \lambda_T \ell_T(\hat{t}, t^*) + \lambda_R \ell_R(\hat{t}, t^*) \quad (1)$$

where $\ell_T(\hat{t}, t^*)$ is the squared Euclidean distance loss:

$$\ell_T(\hat{t}, t^*) = ||\text{Trans}(\hat{t}) - \text{Trans}(t^*)||^2 \quad (2)$$

and $\ell_R(\hat{t}, t^*)$ is the squared deviation of the product of the predicted rotation matrix and the transpose of the ground truth rotation matrix:

$$\ell_R(\hat{t}, t^*) = ||\text{Rot}(\hat{t})\text{Rot}(t^*)^T - I||^2 \quad (3)$$

To train GPNet as used in our experiments, we use weights $\lambda_T = \lambda_R = 1$. GPNet is trained for 2000 epochs on two Nvidia 2080 Ti GPUs for approximately 24 hours using an Adam optimizer with a learning rate of 1×10^{-4} [23].

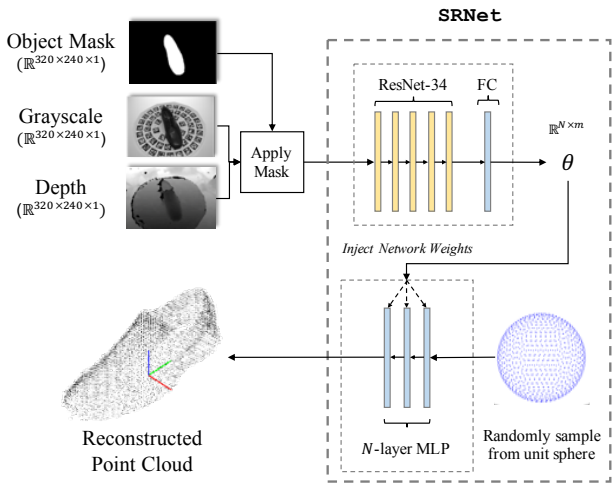

Fig. 4. From foreground-masked grayscale and depth images, our 3D shape reconstruction network, `SRNet`, learns a mapping function f_θ that maps points from a canonical domain such as a unit sphere to a 3D object. Our system leverages the additional geometric information provided by this reconstruction to refine grasps proposed by `GPNet`.

D. 3D Shape Reconstruction Network - `SRNet`

For the 3D reconstruction component of our system, we use the `SRNet` method recently introduced in [18]. As outlined in Figure 4, rather than learning an explicit representation such as a point cloud or an occupancy grid, `SRNet` represents an object as a mapping function f_θ which maps points from a canonical domain S (e.g. a unit sphere in $\mathbb{R}^3$) to the object surface. The mapping function is represented as a Multi Layer Perceptron (MLP) whose weights θ are output from a *Higher Order Function (HOF)*. In other words, the higher order function takes an image as input and outputs the mapping function from S to the object.

Let I be the segmented image of an object and Y^* be the ground truth represented as a set of points sampled from the 3D model of the object. The network is trained as follows: on input I, the higher order function outputs $\theta = g(I)$ which corresponds to the weights of the MLP representing f_θ. Afterwards, a fixed number of points $X = \{x_i\}$ are sampled from the canonical domain uniformly at random. The network output Y is obtained by applying f_θ on each x_i to obtain $Y = \{f(x) : x \in X\}$. For training, the Chamfer distance between Y and Y^* is used as a loss function. The advantage of this formulation is that since the domain is resampled at each iteration, the network can learn representations in arbitrary resolution determined only by the resolution of the ground truth. This representation is comparable to the state of the art in terms of reconstruction accuracy using significantly fewer parameters [18].

To align the generated 3D reconstruction with the perceived object, we use a variant of iterative closest point (ICP) [24] as implemented in Open3D [25]. We assume the shoe to be sitting flat on the table, so we estimate the translation component of the transformation using the centroid of the visible point cloud projected from the depth image. We estimate the rotation component of the transformation by calculating the major axis of the visible point cloud through PCA. This rough estimate provides an adequate initialization for ICP allowing us to precisely align reconstructed point clouds with the observed portion of objects.

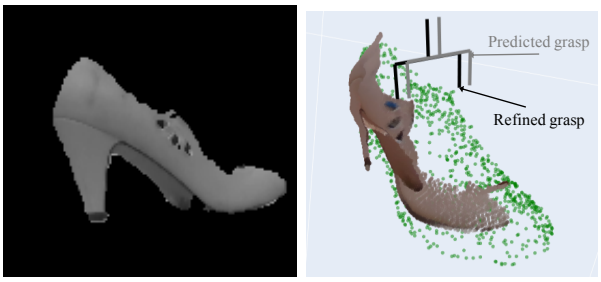

Fig. 5. Projecting the proposed grasp onto the reconstructed point cloud improves grasp accuracy. Left: Grayscale image of shoe. Right: `SRNet` reconstruction overlaid on visible point cloud, with proposed and project grasps.

E. Grasp Refinement And Execution

1) Grasp Refinement: As discussed in Section IV-C.2, `GPNet` produces 6-DOF grasp proposals semantically consistent with the examples upon which it was trained. In the case of the network used in our experiments, this means it will always propose grasps on or near the left side of the shoe, even if only the right side is visible in the image, as in Figure 5. In these cases, `GPNet` must "guess" an appropriate location for the grasp, and its accuracy tends to be lower than when grasping a visible point.

Our system refines the grasp proposed by `GPNet` by projecting it onto the surface of the point cloud reconstruction created by `SRNet`. Given the proposed grasp ${}^{\mathcal{C}}T_{\mathcal{G}}$, we select the point in the point cloud that is closest to the grasp point and refine the grasp by setting its position to the coordinates of that point, leaving its orientation unchanged, resulting in the output grasp ${}^{\mathcal{C}}T_{\mathcal{G}}{}^{+}$. Figure 5 provides an illustrative example. Refining the grasp through this projection operation increases the accuracy of the overall system, and results in better performance in our grasping experiments.

2) Grasp Execution: Our system produces grasp plans in the form of gripper poses with respect to the camera frame. Since we assume the pose of the camera ${}^{\mathcal{R}}T_{\mathcal{C}}$ is known, this camera-frame grasp can be easily transformed into the robot frame: ${}^{\mathcal{R}}T_{\mathcal{G}}{}^{+} = {}^{\mathcal{R}}T_{\mathcal{C}}{}^{\mathcal{C}}T_{\mathcal{G}}{}^{+}$. When executing a grasp on the robot, we command the gripper to first move to an offset point $20cm$ back from the grasp point (along the gripper axis), then move in to the grasp point and close its fingers.

V. Experiments

We perform experiments to characterize the performance of our method at grasping known and unknown shoes, and compare it against alternative systems. All systems tested assume a single shoe placed on a flat table, imaged by a depth camera with a known pose with respect to the table. Shoes are segmented from the background by reprojecting the depth image into a point cloud in the table frame, and then removing all visible points beneath the table plane.

We benchmark the network our full system, ablations, and baselines on the set of four shoes shown in Figure 3. One of them (the Oxford) was withheld from the training data for `GPNet` and `SRNet` as a test set shoe.

A. Systems Tested

We compare our full system against seven alternatives, which we subdivide into *baseline*, *ablation*, and *oracle* systems. Each tested system consists of a grasp proposal and a grasp refinement method. Here we describe these methods, as well as the combinations tested.

1) Grasp Proposal Methods:

`GPNet` (ours) This is the neural net grasp proposer described in Section IV-C.

Naive (baseline) This method proposes an initial grasp 20 cm above the table at the shoe centroid with a uniform random offset of ± 2 cm added to its x and y position (in the table plane) to perturb the grasp off-center before grasp refinement. The gripper is pointed down (normal to the table plane), and its grip axis is aligned to the major axis of the available shoe point cloud. The shoe is assumed to be sitting flat on the table, so shoe position (x, y) is estimated by projecting the centroid (mean) of the visible point cloud down to the table plane, and major axis orientation θ is computed via PCA. This is usually not a reasonable grasp on its own, so its success depends largely on the point on the estimated shoe surface to which it is projected during the refinement step.

Library (oracle) This method proposes the ground-truth training-set grasp ${}^{\mathcal{O}}T_{\mathcal{G}}{}^{*}$ for the shoe being tested. This grasp is defined with respect to the shoe body frame $\mathcal{O}$, whose pose with respect to the camera ${}^{\mathcal{C}}T_{\mathcal{O}}$ is estimated based on the available point cloud; details of pose estimation are provided in the following sections.

2) Grasp Refinement Methods:

`SRNet` (ours) This is the neural net shape reconstruction method described in section IV-D.

Visible (baseline) In this case, no reconstruction is used: grasps are refined by projection onto the visible point cloud from the depth image.

None (baseline) The grasp proposal is used on its own, with no refinement step.

Library (oracle) This method uses the ground-truth point cloud for the shoe being tested, providing an upper bound on the performance of shape reconstruction. As with our 3D shape reconstruction network output, the point cloud is aligned to the observed point cloud via ICP.

3) Combinations Tested:

Full System (ours) Our full system employs `GPNet` for proposal and `SRNet` for refinement, as described in Section IV.

Ablation Systems The ablation systems test the effect of replacing either the proposal or refinement stages of our full system with the baseline alternative. `GPNet` -Visible projects the `GPNet` proposal onto the visible point cloud, and Naive-`SRNet` projects the naive grasp proposal onto the completed point cloud generated by `SRNet`. In the latter case, the position and orientation of the Naive grasp proposal are based off of the alignment of the completed point cloud generated by `SRNet` to the visible point cloud. `GPNet` -None tests the performance of the `GPNet` grasp proposal on its own, with no refinement step.

Baseline Systems The baseline systems use the Visible point cloud for refinement (no shape reconstruction), and the Naive and Library grasp proposals. For the purposes of grasp proposal, the position and orientation of the shoe is estimated based on the centroid and major axis of the visible point cloud, computed via PCA.

Oracle Systems The oracle systems test the effect of replacing either the proposal or refinement stages of our full system with the ground-truth oracle alternative, and represent a upper-bounds on system performance. `GPNet` -Library uses the `GPNet` grasp proposal projected onto the ground-truth point cloud; the ground-truth point cloud is aligned to the visible point cloud using the same technique used for reconstructed point clouds, as described in Section IV-E. Library-`SRNet` projects the ground-truth grasp onto the point cloud reconstructed by `SRNet`; for the purpose of grasp proposal, shoe pose is estimated based on the alignment of the `SRNet` reconstruction to the visible point cloud.

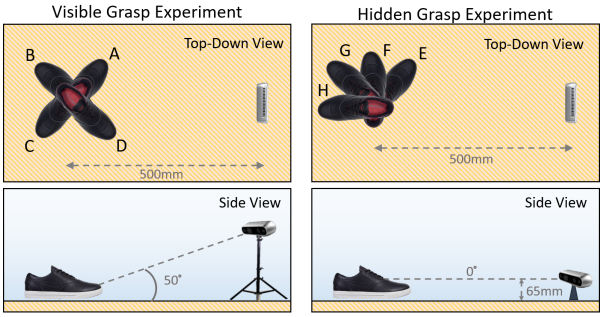

Fig. 6. Visible-grasp (left) and hidden-grasp (right) experimental setups. In the visible-grasp experiment, the camera is $500mm$ away from the shoe at a $50°$ elevation angle, and the shoe is placed in one of four poses (A-D) at $90°$ increments. In the hidden-grasp experiment, the camera is $500mm$ away from the shoe and parallel to the table, while the shoe is placed in one of four poses (E-H) at $45°$ increments. The grasp point on the left side of the shoe is not visible from these viewpoints.

B. Experiment Procedures

1) Experiment 1: Visible-Grasp Shoe Picking Task: This experiment tests the ability of each system to pick up shoes from camera viewpoints that provide an unobstructed view of the grasp point. Each of the four shoes in Figure 3 was tested. The Heel, Pump, and Sneaker were all included in the training set, while the Oxford was withheld as a test set shoe.

Figure 6 shows the experimental setup. In each trial, the shoe is placed on the table, in one of four orientations at $90°$ increments (labelled A, B, C, and D). The camera then positioned to point at the shoe, at a distance of 500 mm and with an elevation angle of $50°$ (from the table plane). A single RGB-D image is captured, segmented, and provided

Type	Proposal	Refinement	Success	Attempts	%
Baseline	Naive	Visible	6	16	38%
Baseline	Library	Visible	8	16	50%
Ablation	Naive	SRNet	6	16	38%
Ablation	GPNet	None	14	16	88%
Ablation	GPNet	Visible	15	16	94%
Full	**GPNet**	**SRNet**	**15**	**16**	**94%**
Oracle	GPNet	Library	14	16	88%
Oracle	Library	SRNet	13	16	81%

TABLE I

VISIBLE-GRASP EXPERIMENT RESULTS

Type	Proposal	Refinement	Success	Attempts	%
Baseline	Naive	Visible	3	16	19%
Baseline	Library	Visible	6	16	38%
Ablation	Naive	SRNet	8	16	50%
Ablation	GPNet	None	12	16	75%
Ablation	GPNet	Visible	9	16	56%
Full	**GPNet**	**SRNet**	**14**	**16**	**88%**
Oracle	GPNet	Library	15	16	94%
Oracle	Library	SRNet	16	16	100%

TABLE II

HIDDEN-GRASP EXPERIMENT RESULTS

as input to the system being tested. The output grasp pose and offset point are then transformed into the robot frame and executed by the robot.

Picking is considered successful if, after closing the gripper and lifting the hand, the shoe is stably grasped and lifted off the table (i.e. it does not move within the grasp under gravity). Additionally, we only consider picking to be successful if the grasp does not damage or deform the shoe. If the body of the shoe is deformed by the grasp when the gripper jaws have closed fully, we consider the grasp to be inappropriate and unsuccessful.

2) Experiment 2: Hidden-Grasp Shoe Picking Task: This experiment tests shoe picking from viewpoints where the desired grasp point is occluded in the visible image. The camera and shoe positioned as shown in Figure 6: the camera is parallel to the table (0° elevation) at a height of 65 mm and a distance of 500 mm to the shoe, which is placed in one of four poses at 45° increments (labelled E, F, G, and H). In each of these poses, the grasp point (on the left side of the shoe) is hidden from view. For example, Figure 5 shows an image of the Pump taken in pose G.

C. Results

Tables I and II show the results for the visible- and hidden-grasp experiments, respectively. Our combined system (Full GPNet-SRNet) performs well at the grasping task under both conditions, with a 94% success rate when the grasp point is visible, and an 88% success rate when the grasp point is hidden. We see a clear performance advantage compared to the baseline systems under both conditions, which have task success rates of under 50%. Comparing the full system to the oracle systems (which represent an upper bound on performance), we see that its performance is comparable in both cases.

Type	Prop.	Ref.	**O**	H	P	S	Total	%
Baseline	Naive	Visible	1/8	4/8	2/8	2/8	9/32	28%
Baseline	Library	Visible	2/8	3/8	7/8	2/8	14/32	44%
Ablation	Naive	SRNet	1/8	5/8	4/8	4/8	14/32	44%
Ablation	GPNet	None	8/8	7/8	6/8	5/8	26/32	81%
Ablation	GPNet	Visible	6/8	6/8	7/8	5/8	24/32	75%
Full	**GPNet**	**SRNet**	**7/8**	**7/8**	**8/8**	**7/8**	**29/32**	**91%**
Oracle	GPNet	Library	7/8	7/8	8/8	7/8	29/32	91%
Oracle	Library	SRNet	8/8	5/8	8/8	8/8	29/32	91%

TABLE III

PER-SHOE GRASPING SUCCESS RATES FROM BOTH VIEWPOINTS. O=OXFORD, H=HEEL, P=PUMP, S=SNEAKER. OXFORD IS A TEST SET SHOE.

In the visible-grasp experiment, the performance of our full system is equal to that of our proposal-only ablation (GPNet-Visible), at a 94% success rate. This makes sense: when the grasp point is visible in the image, refining the grasp proposal by projecting it onto the visible point cloud should provide the same benefit as projecting it onto the reconstructed point cloud. In the hidden-grasp experiment, we see that 3D reconstruction provides a clear advantage: our full system succeeds in 88% of trials, while both ablations (GPNet-Visible and Naive-SRNet) succeed in around 50% of trials.

Table III shows the grasping success rates of all methods broken down by shoe, combining the results from both the hidden- and visible-grasp viewpoints. The full GPNet-SRNet system has an overall success rate of 91%, matching the performance of both oracle systems. It's also worth noting that the 7/8 overall success rate on the Oxford shoe, which was not included in the training set, is comparable to the other shoes that were included in training, indicating the ability to generalize to new objects within a category.

VI. DISCUSSION

Our experimental results illustrate the value of including both a learned grasp proposer and a dedicated 3D-reconstruction component in our architecture. From convenient viewpoints, our grasp proposal model is able to produce successful 6-DOF grasps on its own. However, from challenging viewpoints that hide the desired grasp point, 3D reconstruction boosts performance, allowing our combined system to succeed where a purely pixel-space system would fail.

Employing two convolutional architectures for both the grasp proposal network and 3D reconstruction network allows our system to output accurate 6-DOF grasps without sacrificing the computational advantages of learning over images. Our system operates on relatively high-resolution images (320 x 240), and inference is very fast, about 300 ms for grasp proposition and about 100 ms for 3D reconstruction.

The low success rates of the baselines highlights the difficulty of this precision grasping task: we not only require the robot to lift the shoe off of the table, but to do so in a precise, controlled manner. Frequently, the baseline systems

are able to grasp the shoe, but do so by pinching and deforming the entire body of the shoe, which does not meet our criteria for success. While this kind of grasp may be suitable for durable rigid objects, they are not suitable for deformable, valuable household objects like shoes.

Our system is not without limitations. Alignment of the reconstructed point cloud to the visible point cloud via ICP is time consuming (several seconds in our current implementation), and is sometimes a source of error, especially when very little of the target object is visible, as in pose H of the hidden-grasp experiment (Figure 6). In, particular, alignment is brittle to cases where `SRNet` reconstructs the shoe inaccurately (for example, generating a shape similar to a flat or sneaker in response to a challenging viewpoint of a high heel). In the future, we may explore a learned neural network model to align the visible and reconstructed point clouds, which could result in better performance and faster speed.

Refining grasps by projecting them onto the reconstructed point clouds increases grasping performance. However, there may be utility in performing refinement in a more sophisticated way. Our current refinement method changes the position of the gripper, but leaves its orientation unchanged. In the future, refining the grasp orientation based on the shape of the nearby reconstructed surface could result in better performance - for example, by searching for an orientation where the fingertips are at antipodal points with respect to a flat surface.

VII. Conclusion and Future Work

We presented a grasp planning system that generates accurate 6-DOF grasps from single RGB-D images of an object. Two neural networks process the same input image to produce a 6-DOF grasp proposal and a 3D pointcloud reconstruction. By projecting the grasp onto the nearest point in the pointcloud, our combined system is able to produce more accurate grasps than the image-based grasp proposal network alone. Our experiments demonstrated the efficacy of our system at grasping both known and unknown objects, and ablation studies show the benefit of combining learned grasp proposal with learned 3D reconstruction.

In the future, it would be interesting to explore an implementation of this system where `GPNet` and `SRNet` are combined into a single network with two heads and a shared convolutional encoder. This would increase computational efficiency, and potentially result in better performance as well - training the two networks jointly could allow the loss function to capture the overall system performance (quality of the grasp post-refinement), and allow for the proposition and reconstruction heads to coordinate their outputs for better performance.

VIII. Acknowledgements

We would like to thank Ajinkya Jain for his work as an intern at SAIC-NY, where he implemented the GraspIt pipeline used to generate the example grasps used in this paper.

References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[2] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1–8.

[3] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2901–2910.

[4] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conference on Robot Learning*, 2018, pp. 373–385.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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

[8] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, "Towards reliable grasping and manipulation in household environments," in *Experimental Robotics*. Springer, 2014, pp. 241–252.

[9] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *arXiv preprint arXiv:1903.11239*, 2019.

[10] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[11] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 12 2014.

[12] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European conference on computer vision*. Springer, 2016, pp. 628–644.

[13] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 52–67.

[14] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[15] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[16] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2442–2447.

[17] D. Watkins-Valls, J. Varley, and P. Allen, "Multi-modal geometric learning for grasping and manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7339–7345.

[18] E. Mitchell, S. Engin, V. Isler, and D. D. Lee, "Higher-order function networks for learning composable 3d object representations," in *International Conference on Learning Representations (ICLR)*, 2020, arXiv preprint arXiv:1907.10388.

[19] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," *arXiv preprint arXiv:1903.06684*, 2019.

[20] W. Gao and R. Tedrake, "kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion," *arXiv preprint arXiv:1909.06980*, 2019.

[21] T. Tosun*, E. Mitchell*, B. Eisner, J. Huh, B. Lee, D. Lee, V. Isler, H. S. Seung, and D. Lee, "Pixels to plans: Learning non-prehensile manipulation by imitating a planner," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 7431–7438.

[22] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *Journal of graphics tools*, vol. 8, no. 2, pp. 1–15, 2003.

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

[24] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.

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