2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Real-Time Generative Grasping with Spatio-temporal Sparse Convolution

Timothy R. Player[1], Dongsik Chang[2], Li Fuxin[1], and Geoffrey A. Hollinger[1]

***Abstract*— Robots performing mobile manipulation in unstructured environments must identify grasp affordances quickly and with robustness to perception noise. Yet in domains such as underwater manipulation, where perception noise is severe, computation is constrained, and the environment is dynamic, existing techniques fail. They are too computationally demanding, or too sensitive to noise to allow for closed loop grasping or dynamic replanning, or do not consider 6-DOF grasps. We present a novel grasp synthesis network, TSGrasp, that uses spatio-temporal sparse convolution to process a streaming point cloud in real time. The network generates 6-DOF grasps at greater speed and with less memory than Contact GraspNet, a state-of-the-art algorithm based on PointNet++. By considering information from multiple successive frames of depth video, TSGrasp boosts robustness to noise or temporary self-occlusion and allows more grasps to be rapidly identified. Our grasp synthesis system was successfully demonstrated in an underwater environment with a Blueprint Labs Bravo robotic arm.**

I. Introduction

Robotic grasp synthesis on unseen objects is a crucial skill that enables the application of generalized robotics to manufacturing, home service, and scientific applications. While state-of-the-art robotic grasping systems achieve reliable performance when grasping previously unseen objects in laboratory conditions, robots deployed in real-world environments must continue to operate when visual conditions are degraded or when environments are dynamic.

In the underwater grasping domain, visual obscurants, adverse lighting, and energetic disturbances require that grasp synthesis algorithms accommodate adverse visual conditions and provide real-time updates suitable for closed-loop control or dynamic replanning. Motivated by the problems of underwater sample collection [1] and infrastructure maintenance [2], we present a novel technique for quickly finding 6-DOF grasp poses in streaming depth video. The proposed system combines information from consecutive depth images obtained from a moving depth camera, to produce a diverse set of stable grasp poses. To do this, the algorithm uses spatio-temporal sparse convolution, a technique for efficiently processing signals by convolving across both spatial and temporal dimensions that has been previously applied to point cloud classification [3], object tracking [4], and lidar video segmentation [5].

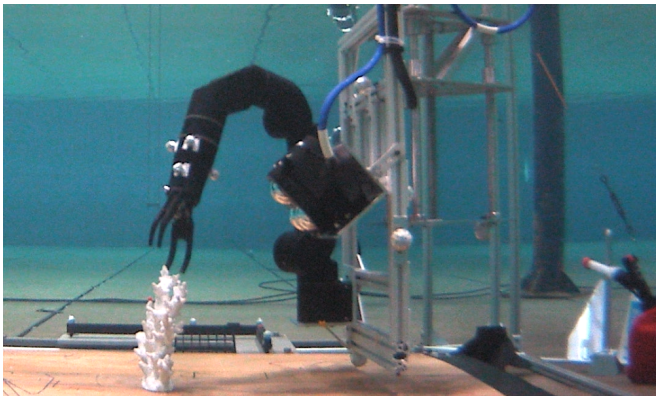

Fig. 1. Successful grasp of a plaster coral branch underwater in the O.H. Hinsdale Wave Lab using the proposed real-time grasp synthesis algorithm with spatio-temporal sparse convolution.

To our knowledge, our work is the first to use information from multiple camera frames to generate dense 6-DOF candidate grasps at sufficient speed for closed loop grasping. Our primary contribution is a learning formulation for 6-DOF grasp inference from depth video using spatio-temporal convolution. Comprehensive results indicate the suitability of our approach for closed loop grasping, and we provide a successful demonstration of grasping unseen objects underwater in the O.H. Hinsdale Wave Research Laboratory at Oregon State University, USA.

Results in simulation indicate that, by reducing the memory and computational requirements relative to PointNet++ [6], spatio-temporal convolution improves the coverage metric [7], retrieving a greater proportion of the grasps in a scene by classifying more points without sacrificing the accuracy of the grasp hypotheses. Simulated results also suggest that basing inference on multiple frames in a moving trajectory can improve precision and recall of grasps. Results on real camera sensors compare the performance of the algorithm in two settings: on point clouds from a structured-light Intel Realsense camera above ground, and on point clouds from a noisier underwater stereo camera underwater called the Trisect [8]. We find that our grasp classification algorithm using spatio-temporal networks improves consistency between the results from high- and low-quality point clouds.

II. Related Work

Grasp synthesis is an important problem in robotics that has been studied for decades. While classical methods

[1] These authors are with the Collaborative Robotics and Intelligent Systems Institute, Oregon State University, Corvallis, OR 97331, USA. {playert, fuxin.li, geoff.hollinger}@oregonstate.edu.

[2] This author was with the Collaborative Robotics and Intelligent Systems Institute, Oregon State University, Corvallis, OR 97331, USA. dongsikc3@gmail.com.

This work was supported in part by ONR award N0014-21-1-2052 and ONR/NAVSEA contract N00024-10-D-6318/DO#N0002420F8705 (Task 2: Fundamental Research in Autonomous Subsea Robotic Manipulation).

leverage geometric and physical principles to analytically determine viable grasps [9]–[12], more recent approaches use machine learning approaches to classify or generate grasps after training on real or simulated data [13]–[15]. Like these, our work adopts a supervised learning formulation to generate grasps from training data including labeled instances of positive and negative grasps.

Within the set of learning-based approaches, recent advances in deep learning have enabled fast generation or evaluation of feasible 2D or 6-DOF candidate grasps. One body of work focuses on 2D "planar" grasp generation, in which grasps are assumed to be executed from a top-down direction by a robot with a two fingered gripper in a bin-picking application. Because these grasps have few degrees of freedom, they can be robustly learned by smaller networks from smaller datasets, and can be inferred at a faster rate [13], [16]. However, 2D planar grasping algorithms do not consider grasps executed from another direction than the focal axis. As a result, potentially suitable grasps are overlooked. 6-DOF grasping algorithms consider the three-dimensional shape of the object in order to reason about grasps executed from arbitrary gripper poses in $\mathrm{SE}(3)$. 6-DOF grasping algorithms have been shown to be successful in cluttered environments [17], [18], where more collision-free grasps can be found with additional freedom in approach direction. In either 2D or 6-DOF grasping, algorithms may be classified as generative–producing a dense, pixelwise estimate of grasp quality–or discriminative, providing the ability to evaluate arbitrary grasp hypotheses. For our grasp synthesis algorithm, we have chosen a generative formulation similar to [16] and [7]. However, in contrast to these works our network uses sparse convolution to generate grasps at greater speed.

Many robotic grasping approaches separate grasp synthesis from control by executing the single most-confident grasp without re-evaluating grasps during execution, which can lead to failure when the best grasp is initially occluded [19]. To address this shortcoming, recent work in closed-loop grasping has combined visual servoing with online grasp synthesis. Approaches such as [16] and [20] repeatedly generate 2D planar grasp candidates, and execute trajectories based on the latest identified best grasp. Similarly, approaches from reinforcement learning seek to learn behavior policies that identify actions leading to secure grasps in closed loop [21]. Within closed-loop grasping, recurrent Bayesian state estimation has been shown to help identify occluded grasps in cluttered scenes [20], [22]. These techniques have combined 2D planar grasp hypotheses obtained from different perspectives to determine the most likely grasp target. However, these closed-loop grasp techniques rely on low latency, high-frequency inferences of grasps in the scene; our work meets this need by generating 6-DOF grasps in real time.

In robotic grasping environments where the camera has limited maneuverability, 2D planar grasp generation may fail to identify viable grasps. In the context of underwater grasping–where the environment is dynamic, visibility is variable, and the camera is moving–the need for real-time generation of 6-DOF grasps using images from multiple instants in time is keenly felt. Inspired by the stated advances in 6-DOF grasp generation and closed-loop grasping, our algorithm generates 6-DOF grasps online from sequences of depth images, using spatio-temporal sparse convolution to incorporate information from multiple images. In contrast to related work, we show that spatio-temporal convolution accounts for shifting occlusions more effectively than repeated single-view grasp generation, while providing a sufficiently high update rate for robust closed-loop grasping in dynamic environments.

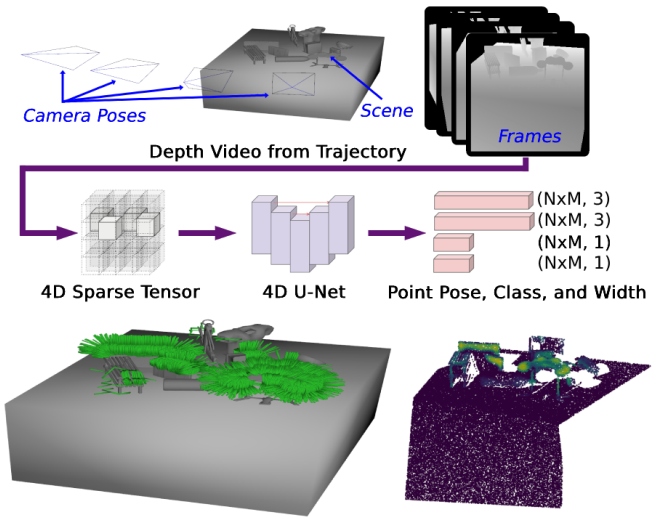

Fig. 2. Proposed multi-frame grasp synthesis system. Top: successive frames of a depth video are recorded throughout the camera's trajectory. Middle: a queue of multiple frames is collated into a four-dimensional tensor describing an occupancy map with spatial and temporal coordinates. A 4D U-Net processes the tensor with sparse convolution to infer grasp confidence and gripper approach direction, baseline direction, and width for each point. Bottom: identified grasps (left) and point confidences (right, lighter color is more confident) for example scene.

III. Method

Our proposed spatio-temporal grasp synthesis method, called TSGrasp, generates a dense classification of points in a streaming depth video to identify a 6-DOF grasp pose corresponding to each point, while optionally fusing information from multiple consecutive depth video frames. Similar to Contact GraspNet [7] and GGCNN [16], we separate grasp synthesis into pointwise classification and regression of the gripper orientation. However, unlike Contact GraspNet and GGCNN, to generate grasps, we apply convolutional filters to a sparse tensor whose entries are the occupancy values of a voxel grid representing the spatial X, Y, Z, and time T coordinates of a point cloud sequence.

A. Problem Formulation

We target the 6-DOF grasp synthesis problem, the goal of which is to find gripper configurations within a scene, consisting of a gripper pose $g \in \mathrm{SE}(3)$ and gripper width w, that would stably secure an object in the jaws for manipulation.

B. Classification and Regression Representation

To solve this problem, we apply supervised deep learning in a manner inspired by Contact GraspNet [7], by training a deep network to classify input points by proximity to stable grasps and regressing the corresponding gripper orientations. Rather than directly learning a grasp quality function $f : \mathrm{SE}(3) \to \mathbb{R}$ to evaluate arbitrary poses and identify stable grasp poses in a scene, we consider only those grasp poses for which the gripper's fingers contact objects within the scene. For each input point, we infer a gripper orientation R_g, gripper width w, and confidence value c that fully define the pose and predicted quality of a grasp for a given point. This learning formulation generates a dense set of candidate grasps in a single shot without requiring optimization over a quality function as discriminative methods do, ensuring fast execution and broad coverage of the scene.

We represent a grasp pose g by a homogeneous matrix

$$g = \begin{bmatrix} R_g & \mathbf{t}_g \\ 0 & 1 \end{bmatrix}, \tag{1}$$

with the grasp's origin $\mathbf{t}_g$ translated from a point $\mathbf{p}$ in the input point cloud by a vector depending on the gripper's baseline direction $\mathbf{b}$, approach direction $\mathbf{a}$, and grasp width w by

$$\mathbf{t}_g = \mathbf{p} + \frac{w}{2}\mathbf{b} + d\,\mathbf{a}. \tag{2}$$

We refer the reader to [7] for an illustration of the relevant parameters on a robotic gripper. With $\times$ denoting the cross product, the grasp's orientation is given by

$$R_g = \begin{bmatrix} | & | & | \\ \mathbf{b} & \mathbf{a} \times \mathbf{b} & \mathbf{a} \\ | & | & | \end{bmatrix}. \tag{3}$$

By directly regressing $\mathbf{a}$, $\mathbf{b}$, w, and inferring the class confidence c, we avoid the costly problem of estimating quality of all grasps in $\mathrm{SE}(3)$ by instead classifying points and regressing orientations in a one-shot manner. As in [7], the vectors $\mathbf{a}$ and $\mathbf{b}$ defining the gripper orientation are normalized to form orthonormal columns of the rotation matrix in Eq. (3), which is a learnable parameterization of $\mathrm{SO}(3)$ [23].

C. Spatio-temporal Network

In contrast with existing methods that use PointNet++ to process point clouds from a single depth image [7], [17], [24], [25], our network uses spatio-temporal sparse convolution to identify grasps. This allows us to process multiple point clouds at once, fusing information from multiple frames and potentially multiple perspectives. If the camera is moving, combining multiple frames during inference allows the network to consider points lying on surfaces of a scene that are occluded from a single perspective; if the camera is not moving, then information from multiple frames can still boost robustness to instantaneous noise. The grasp synthesis module is input a sequence $\mathcal{P}$ of N point clouds with M points $\mathbf{p}$, such that

$$\mathcal{P} = \{\mathcal{P}_i\} = \{\mathbf{p}_{i,j}\}, \tag{4}$$

where $i = 1, \cdots, N$ and $j = 1, \cdots, M$. This sequence of point clouds is obtained from consecutive frames of a depth video (transformed into $\mathbb{R}^3$ via camera intrinsics), as shown in Fig. 2. The resulting grasp predictions are based on a sliding window of previous observations throughout the camera's trajectory. If the camera is moving, information from different perspectives reduces the effect of self-occlusion in the scene, potentially clarifying grasp hypotheses that had been uncertain.

The N-frame trajectories are known. While odometry noise can accumulate during underwater operations, the accumulation of this noise is limited because we require only the relative odometry during the N-frame sequence, in contrast to approaches requiring point clouds to be in a global frame. Using the relative camera poses $T_{\text{world,cam},i}$, the point clouds are transformed into the reference frame of the most recent camera pose (at frame N):

$$\mathcal{P}_{\text{latest},i} = (T_{\text{world,cam},N})^{-1} T_{\text{world,cam},i}\, \mathcal{P}_i. \tag{5}$$

The sequence $\mathcal{P}_{\text{latest},i}$ is discretized into a uniform voxel grid and represented as a sparse tensor with coordinates and features

$$C = \begin{bmatrix} x_1 & y_1 & z_1 & t_1 \\ & \vdots & & \\ x_{NM} & y_{NM} & z_{NM} & t_{NM} \end{bmatrix}, \quad F = \begin{bmatrix} f_1 \\ \vdots \\ f_{NM} \end{bmatrix}, \tag{6}$$

where each coordinate gives the voxel location and timestamp of a single point from the point cloud sequence, and each feature gives the number of points in the corresponding voxel. The timestamp coordinates are integers $0, \ldots, N-1$. We process this representation using sparse convolution with the Minkowski Engine backend [3], which is a fast and memory-efficient way to learn features from sparse spatio-temporal data such as depth video. We use a Minkowski Engine UNet14 backbone [3] to extract features from the input points, which are then processed through separate convolutional network heads to produce the four outputs for each input point. The four outputs include the point class likelihood c (a scalar), the gripper approach direction $\mathbf{a} \in \mathrm{R}^3$, the gripper baseline direction $\mathbf{b} \in \mathrm{R}^3$, and the gripper width w.

D. Data Generation and Training

We train our network using the ACRONYM dataset [26], which consists of $8,872$ digital object meshes labeled with the poses of 17.7M successful and unsuccessful parallel-jaw grasps.

First, we produce cluttered tabletop scenes using the scene generation pipeline from [7]. Then, we transform the positive grasp poses and gripper contact points from the dataset, which are initially expressed in the object frame, into the camera frame at each pose along the trajectory. Then, we render depth images of these scenes from consecutive perspectives along random trajectories of length N. We label input points from the depth video as successful if they are sufficiently close to a gripper contact point from a successful grasp.

Specifically, we generate a trajectory of N camera poses viewing each tabletop scene. With each trajectory, the camera orbits the center of the scene at a fixed distance d and azimuth angle ϕ, while varying the yaw θ at a constant rate $\Delta\theta$. For each example of a tabletop scene seen during training, the distance, azimuth, and initial yaw θ_0 are drawn uniformly, while $\Delta\theta$ is drawn from a normal distribution. With the notation $\mathcal{U}(x_{\min}, x_{\max})$ to denote a uniform distribution and $\mathcal{N}(\mu, \sigma)$ to denote a normal distribution, we used the parameters

$$d \sim \mathcal{U}(1.5\,\mathrm{m}, 2.5\,\mathrm{m}), \qquad \phi \sim \mathcal{U}(20°, 90°), \tag{7}$$

$$\theta_0 \sim \mathcal{U}(0, 2\pi), \qquad \Delta\theta \sim \mathcal{N}(0°, 1°), \tag{8}$$

selected to generate diverse orbital trajectories with moderate distance and low tangential velocities (likely during visual exploration of an underwater object by an AUV).

The generated camera poses have yaw

$$\theta_i = \theta_0 + i\,\Delta\theta, \tag{9}$$

where $i = 1, \ldots, N$. The poses are represented as homogeneous matrices $T_{cam,i}$. Because the ACRONYM dataset contains the 6-DOF poses of stable grasps $g \in \mathrm{SE}(3)$ in the frame of each object, these grasp poses are transformed into the camera frame with

$$g_{cam,i} = (T_{cam,i})^{-1}\, g_{obj,i}, \tag{10}$$

and the corresponding gripper contact points $\{\mathbf{c}\}$ are transformed into the camera frame with

$$\mathbf{c}_{cam,i} = (T_{cam,i})^{-1}\, \mathbf{c}_{obj,i}. \tag{11}$$

We denote the video sequence of N point clouds produced by a simulated depth camera as $\{\mathcal{P}_i\}$. Each instantaneous point cloud has M points, such that $\mathcal{P}_i = \{\mathbf{p}_{i,j}\}$, $j = 1, \cdots, M$. Each point is classified by its proximity to positive gripper contact points from ground truth. For all $i = 1, \ldots, N$ and $j = 1, \ldots, M$,

$$s_{i,j} = \begin{cases} 1 & \min_k \|\mathbf{p}_{i,j} - \mathbf{c}_{i,k}\|_2 < r, \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where each vector $\mathbf{c}_{i,k}$ contains the coordinates of gripper contact points in the camera frame in image i and $r \in \mathbb{R}$ is the threshold radius for labeling points. Thus, the points from the depth video may be partitioned into $\mathcal{P}_i^- := \{\mathbf{p}_{i,j} | s_{i,j} = 0\}$ and $\mathcal{P}_i^+ := \{\mathbf{p}_{i,j} | s_{i,j} = 1\}$, depending on whether a suitable grasp has been found near to each input point. The ground truth point classifications, as well as the unique nearest grasp pose corresponding to each contact point, are passed to the network during training to assess losses.

The network is trained using separate loss terms for the point class, gripper pose, and width. We refer the reader to [7] for formal definitions of the loss terms $l_{bce,k}$, l_{add-s}, and l_{width}, which refer to the binary cross entropy classification loss, gripper pose loss, and binned width. The total loss is $l = \alpha l_{bce,k} + \beta l_{add-s} + \gamma l_{width}$, with $\alpha = 1, \beta = 10, \gamma = 1$ (same values as [7]). During training, the average loss is computed for points at all time coordinates. We used trajectory lengths of $N = 1$ and $N = 4$, using success radius $r = 0.005$ m (from [7]), and a spatial discretization bin width of 0.005 m. Networks were trained for 100 epochs using the Adam optimizer with a learning rate of 0.00025 and exponential learning rate decay of 0.99.

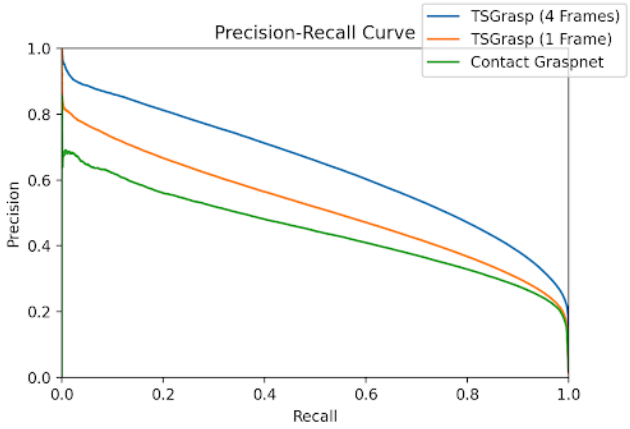

Fig. 3. Precision-recall curves on the test partition of the ACRONYM dataset after labeling input points by proximity to the points at which the gripper contacted objects in the scene for ground truth grasps with a 0.005 m threshold. Precision is the proportion of positively classified points that had positive labels. Recall is the proportion of positively labeled points that were positively classified. The acceptance threshold was varied from 0 to 1 to produce these curves.

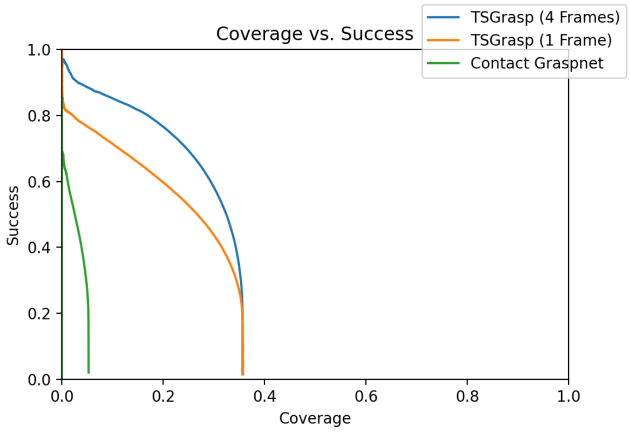

Fig. 4. Success-coverage curve on the test partition of the ACRONYM dataset after labeling input points as in Fig. 3. Success is the proportion of positively classified points that had positive labels. Coverage is the proportion of ground-truth contact points that had positively classified-input points within 0.005 m. The acceptance threshold was varied from 0 to 1 to produce these curves.

IV. Results

The grasp synthesis algorithm was tested in on the simulated ACRONYM dataset and validated during underwater grasp executions at the O.H. Hinsdale Wave Research Laboratory at Oregon State University using a Blueprint Labs Bravo 7 robotics arm and a Trisect underwater stereo camera.

A. Simulation Results

Inference performance was evaluated on the test partition of the ACRONYM dataset. 2,000 cluttered tabletop scenes were generated containing different objects than were

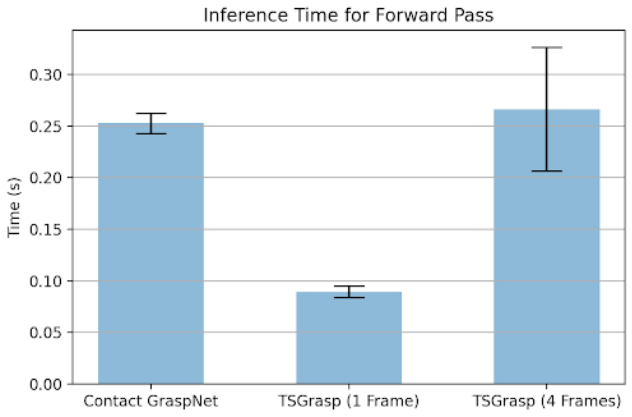

Fig. 5. Inference time for Contact GraspNet versus proposed algorithm (TSGrasp) using 1 frame and 4 frames.

present in the training set, using the same randomly allocated test/train object partitions from [7]. On each scene, trajectories were generated using Eq. (9). Depth videos were rendered along each trajectory, and the corresponding ground truth annotations were transformed into the camera frame as described in Eqs. (11) and (10). Ground truth labels were generated for each point in the depth video using Eq. (12).

For each video sequence, the performance of three networks was compared: the first was TSGrasp trained on trajectories of length $N = 4$, referred to as TSGrasp (4 Frames). During test inference, a sliding-window queue of four frames was input into TSGrasp (4 Frames). This assesses the effectiveness of multi-frame inference. The second network was TSGrasp trained on single frames (trajectories of length $N = 1$). To process the test videos, each video frame was passed in consecutively rather than four at a time. Compared to Contact GraspNet, this assesses the effectiveness of the sparse convolution backbone. Contact Graspnet, without object segmentation, was also tested on single frames from the video sequence.

The precision and recall of point classification were computed for different confidence thresholds, and the resulting precision-recall curve is shown in Fig. 3. TSGrasp (1 Frame) exhibits a higher area under the curve than Contact Graspnet. TSGrasp is based on a Minkowski Engine sparse convolutional backbone that has been shown to outperform PointNet++ in certain classification tasks [3], so the improved classification performance may be partially attributed to the greater expressiveness of the learned backbone features. However, improved precision may also be attributed to the greater number of points that can be efficiently processed by sparse convolution; because the points are discretized into a sparse tensor, TSGrasp is able to retain 45,000 input points in its initial layers compared to 2,048 in Contact GraspNet which is based on PointNet++. Additional input points may better capture local geometry than the more severely downsampled input.

TSGrasp (4 Frames) further outperforms TSGrasp (1 Frame) and Contact GraspNet, exhibiting higher area under the PR curve. During inference, four consecutive frames of an orbital trajectory (Eq. (9)) are processed. The additional perspectives from different angles tend to capture a greater proportion of the scene's surfaces than is possible from a single perspective. Consequently, classification performance is improved as grasp hypotheses which may erroneously be labeled as positive based on an partial point cloud can be more effectively eliminated; inference is based on a more complete object model without requiring a slow and costly explicit reconstruction of the object.

As seen in Fig. 4, performance on the test set also demonstrates that temporal convolution with multiple frames can improve the *success* and *coverage* of grasp classification, metrics adapted from [17]. Here, coverage refers to the proportion of positively-labeled contact points from the ground truth data set with a positively-inferred point near them (within 0.005 m). Success refers to the proportion of positively-inferred contact points with a positively-labeled contact point near it. Coverage has a theoretical upper bound because, in a given scene, some of the ground-truth contact points will have no points from the input depth image near them due to self-occlusion within the scene. Both TSGrasp (1 Frame) and TSGrasp (4 Frame) exceed the maximum coverage possible with a single pass of Contact GraspNet without object segmentation. This is because many more points from the input point cloud are classified, resulting in a greater proportion of ground truth positive contact points being "covered". Still, processing additional frames with a moving camera perspective improves success and coverage of the point cloud as the confidence threshold is varied.

Inference time on the test set (N=2,000) with a desktop computer with an RTX 2060 graphics card and Ryzen 5 3600 CPU was faster with sparse convolution in the single-frame case than with the PointNet++ based Contact Graspnet. The version of TSGrasp inferring grasps within a single frame performed inference much faster (0.089$\pm$0.005 s) than Contact Graspnet (0.253$\pm$0.010 s), with the version of TSGrasp processing a queue of four frames approximately linearly increasing inference time relative to single-frame (0.266$\pm$0.060 s). The inference time of TSGrasp is dominated by the time taken to discretize and hash the input point cloud into a sparse tensor for processing on the GPU, while for Contact GraspNet most time is spent within the feature propagation and set abstraction layers of the PointNet++ backbone [6]. Increasing the number of frames used for inference may increase the standard deviation of inference time due to increased variety in the number of discrete 4D bins requiring hashing under different camera trajectories. The dramatic speedup afforded by sparse convolution enables many more points to be processed at a faster speed, or for multiple frames to be processed at approximately the same speed as a PointNet++ based backbone. Minimal inference time is essential for dynamic replanning and closed-loop control in energetic environments such as underwater.

B. Real-World Results on Underwater Testbed

We deployed the grasp synthesis system on a real robotic gantry testbed in an indoor pool environment at the O.H. Hinsdale Wave Research Laboratory. Modern RGB-D cam-

TABLE I
CONSISTENCY IN LAB AND UNDERWATER GRASP CLASSIFICATION

	TSGrasp (4 Frames)	TSGrasp (1 Frame)	Contact GraspNet
RMSE	0.055	0.076	0.092

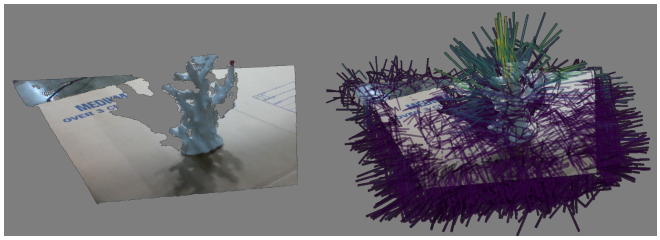

Fig. 6. Input image (left) and inferred grasps (right) from grasp synthesis on the Intel Realsense D435i camera in laboratory setting. Grasps are colored by confidence: purple low, yellow high.

eras such as the Intel Realsense [27] often use structured light sensors to produce high-quality point cloud; in the underwater environment, structured light sensors have not been developed. Our deployment scenario involved a novel underwater stereo camera under development at the Applied Research Laboratory at the University of Washington, called the Trisect [8]. As shown in Fig. 1, a Blueprint Labs Bravo 7 robotic arm [28] and the stereo camera were mounted on a moveable aluminum gantry. Objects could be placed in front of the camera. A limited number of repeatable trials were possible due to the time and cost constraints associated with testing in this facility. We were able to achieve an 80% success rate (4/5 successful grasps) on our main test object, an artificial coral shown in Figure 1. We also achieved several successful grasps on other objects, including a metal corner bracket and a gas can with handle. Successful grasps for all three objects are shown in the accompanying video. The main source of grasp failure stemmed from inconsistent point clouds generated by the Trisect camera, which motivates additional research on filtering and point cloud generation for the underwater camera. Examples of grasps failing in this way are provided in the accompanying video.

Robustness to Underwater Noise in Point Cloud: The point cloud from the underwater camera was subject to both spatial and temporal noise. Spatial noise includes artifacts at the boundaries of objects where the stereo disparity algorithm misidentifies outlier points that should be empty space, and warping of the point cloud attributed to miscalibration of the individual monocular cameras in the stereo system. Temporal noise is attributed to motion in the scene affecting the environment, such as rippling reflections causing areas of light and dark on the floor of the pool. These can lead to inconsistent estimates of the depth of pixels within the depth image, or a momentary loss of correspondence in the disparity map that produces the depth image.

Due to the difficulty of creating ground truth data in underwater scenarios, we cannot evaluate measures of accuracy on the underwater data. We propose to instead measure the consistency of the grasp synthesis algorithms w.r.t. the same scene captured with the high-quality Realsense compared to the noisier underwater stereo camera. Realsense point cloud quality is more similar to the synthetic data in which TSGrasp shows significant improvement over prior work, hence consistency w.r.t. results on the Realsense camera should indirectly indicate the robustness and accuracy of the algorithm. We reproduce the underwater scene in a laboratory setting, producing a similar point cloud with nearly identical positioning (position uncertainty $\approx$0.005 m, rotation uncertainty $\approx 2^{\circ}$). We compared the grasp classifications generated from the RealSense camera with the grasps generated from the Trisect stereo camera. We partitioned the grasp poses into discretized bins of width 0.03 m based on position, obtained the arithmetic mean of the confidence of each bin, and determined the RMSE of all the bin confidences between the RealSense grasps and the Trisect grasps for each algorithm. Table I gives this comparison metric.

TSGrasp (4 Frames) has lower RMS error than TSGrasp (1 Frame) or Contact GraspNet, indicating that results were more consistent between the laboratory environment and the underwater camera, which may indicate greater robustness to noise, potentially due to learned filtering from temporal convolution enforcing temporal consistency to overcome noise [3]. TSGrasp (1 Frame) has lower RMS than Contact GraspNet, which may be due to sparse convolution being less sensitive to noise out of the training distribution than PointNet++. These results indicate that temporal convolution may boost robustness to time-varying noise.

V. CONCLUSION

We presented a novel technique for 6-DOF grasp synthesis that can produce grasp estimates in real time while processing multiple frames at a time. We demonstrated that sparse convolution can achieve state-of-the-art grasp classification results while improving inference efficiency. Alternatively, by increasing the number of frames being processed, grasp classification performance can be boosted at the cost of greater inference time. We successfully implemented our technique on a robotic arm in an underwater environment, which successfully grasped three diverse objects.

These outcomes are particularly useful in the context of mobile manipulation, where operation in unstructured environments such as energetic underwater scenes motivates the need for visual servoing, closed loop grasping, and dynamic replanning. By fusing information from multiple frames without requiring explicit object reconstruction, we avoid CPU-intensive reconstruction. The resulting speed improvements could be used to improve reactive mobile manipulation in dynamic environments. Additional investigation of the tradeoffs encountered when increasing the number of frames used for inference would aid this effort. Furthermore, active grasp exploration, where the agent intentionally maneuvers to perceive the environment from a different perspective to improve grasp success rate [29], [30], may particularly benefit from the multi-frame fusion afforded by spatio-temporal convolution.

REFERENCES

[1] D. Chang, S. Chow, T. Player, and G. A. Hollinger, "Adaptive and informative planning for an underwater vehicle-manipulator system,"

in *IEEE International Conference on Robotics and Automation 1st Advanced Marine Robotics TC Workshop: Active Perception*, Xi'an, China, 2021.

[2] E. Zereik, M. Bibuli, N. Mišković, P. Ridao, and A. Pascoal, "Challenges and future trends in marine robotics," *Annual Reviews in Control*, vol. 46, pp. 350–368, 2018.

[3] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[4] J. Gwak, S. Savarese, and J. Bohg, "Minkowski tracker: A sparse spatio-temporal r-cnn for joint object detection and tracking," *arXiv preprint arXiv:2208.10056*, 2022.

[5] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, "Receding moving object segmentation in 3D lidar data using sparse 4d convolutions," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7503–7510, 2022.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-DOF grasp generation in cluttered scenes," in *Proc. IEEE International Conference on Robotics and Automation*, 2021, pp. 13 438–13 444.

[8] "Trisect," [Accessed Sep. 9, 2022]. [Online]. Available: https://trisect-perception-sensor.gitlab.io/

[9] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 152–165, 1993.

[10] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE International Conference on Robotics and Automation*, 2000, pp. 348–353.

[11] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[12] D. Berenson, R. Diankov, Koichi Nishiwaki, Satoshi Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *Proc. IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 42–48.

[13] 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:1703.09312 [cs]*, 2017.

[14] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[15] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, p. 289–309, 2014.

[16] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *Robotics: Science and Systems XIV*, pp. 1–10, 2018.

[17] A. Mousavian, C. Eppner, and D. Fox, "6-DOF GraspNet: variational grasp generation for object manipulation," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[18] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[19] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[20] H. Schaub, A. Schöttl, and M. Hoh, "6-DOF grasp detection for unknown objects using surface reconstruction," in *Proc. 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, 2021, pp. 1–6.

[21] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[22] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *Proc. IEEE International Conference on Robotics and Automation*, 2019, pp. 8762–8768.

[23] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[24] Y. Li, T. Kong, R. Chu, Y. Li, P. Wang, and L. Li, "Simultaneous semantic and collision learning for 6-DOF grasp pose estimation," *arXiv:2108.02425 [cs]*, 2021.

[25] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 444–11 453.

[26] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *Proc. IEEE International Conference on Robotics and Automation*, 2021, pp. 6222–6227.

[27] Depth camera d435i. [Accessed Sep. 10, 2022]. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/

[28] Reach bravo: Larger platform ROV manipulator - REACH ROBOTICS. [Accessed Sep. 10, 2022]. [Online]. Available: https://reachrobotics.com/products/manipulators/reach-bravo/

[29] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[30] E. Arruda, J. Wyatt, and M. Kopicki, "Active vision for dexterous grasping of novel objects," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2881–2888.