Multiple View Performers for Shape Completion

David Watkins-Valls\textsuperscript{1}, Peter Allen\textsuperscript{1}, Krzysztof Choromanski\textsuperscript{2}, Jacob Varley\textsuperscript{2}, and Nicholas Waytowich\textsuperscript{3}

Abstract—We propose the Multiple View Performer (MVP) - a new architecture for 3D shape completion from a series of temporally sequential views. MVP accomplishes this task by using linear-attention Transformers called Performers [1]. Our model allows the current observation of the scene to attend to the previous ones for more accurate infilling. The history of past observations is compressed via the compact associative memory approximating modern continuous Hopfield memory, but crucially of size independent from the history length. We compare our model with several baselines for shape completion over time, demonstrating the generalization gains that MVP provides. To the best of our knowledge, MVP is the first multiple view voxel reconstruction method that does not require registration of multiple depth views and the first causal Transformer based model for 3D shape completion.

I. INTRODUCTION

Shape completion from a single image or two images is a difficult and important problem (see: [2, 3, 4, 5, 6, 7, 8, 9]). Providing more accurate reconstructions of objects helps enable a variety of robotic tasks such as manipulation, collision checking, sorting, and cataloging. Synthesizing multiple images for accurate 3D-reconstruction of an object is even more challenging.

Fig. 1: Shown in red are partial views of the target object and in green are prediction of each incremental view using our Performer-based MVP to shape completion. The final prediction, in the top right, is the culmination of multiple successive views contributing to an overall completion of the target object, in this case a drill from the YCB object dataset [10].

We propose a novel approach to 3D shape reconstruction, called Multiple View Performer (or: MVP) that can be used to complete objects with only two views, or up to an arbitrary number of views, leveraging recently introduced class of scalable linear-attention Transformers [11], called Performers [1, 12]. At MVP runtime, 2.5D views about the object or simple scene are captured in a panning motion to create a sweeping snapshot of the object’s geometry (see Figure 1), or from a still camera in the case of moving objects. For each of these views, the causal performer block updates its corresponding compact associative memory (approximating modern continuous Hopfield memory [13]), effectively improving MVP’s understanding of an object and consequently - the overall shape estimation. Crucially, the size of the aforementioned compact associative memory is independent from the number of views it consumed (see: Section III-A for more details). When a completion is requested, the current observation implicitly interacts with all the previous observations through that compact memory for its more accurate infilling. Due to the Performer block’s ability to memorize multiple views, it can also remember objects that are no longer visible or utilize newly revealed views of objects that were previously hidden. Through our results, we will show that the proposed MVP system is able to generalize better both for single view and multiple view reconstruction without requiring the registration of multiple views of the object. We will show that this shape completion system is able to perform better or on par versus an LSTM-based system and an attention-based system. This system can be used for many different robotics tasks, such as grasping, stacking, and collision avoidance. We show a real world demonstration of how this could be used with a camera fixed to a robot in Figure 2. We also demonstrate, using a simulated BarrettHand, that this shape completion system can be used for grasp planning.
II. RELATED WORK

A single-depth sensor can be moved to capture multiple views of an object, but aligning those views remains hard. Work done by Tremblay et al. [14] has shown that a grasp planner can plan and execute a grasping task given a mesh and an estimated pose of that mesh. Other work from Varley et al. [8] and Watkins-Valls et al. [15] have shown that a single view or two views of an object can be used to reconstruct the object for grasp planning via GraspIt! [16]. Other work by Wang et al. [17] has shown that objects can be manipulated via dense fusion of multiple images of the object by requiring registration between those images.

There has been much successful research in utilizing continuous streams of visual information like Kinect Fusion [18] or SLAM [19] to improve models of 3D objects for manipulation, an example being [20, 21]. A popular approach (see for instance RTABMap [22]) is to perform visual RGBD slam to register multiple point clouds. It can be challenging for a robotic agent to denoise registered images and eliminate potentially introduced errors into a machine-learned CNN. This type of noise is difficult to model in simulation. A 3D-CNN [23] can be used to enable robust shape estimation by leveraging multiple unregistered view, meaning that each image of an object is kept in its respective image frame.

Zha et al. [24] developed an approach to build 3D models of unknown objects based on a depth camera observing the robot’s hand while moving an object. This strategy integrates both shape and appearance information into an articulated ICP approach to track the robot’s manipulator and the object while improving its 3D model. Similarly, another work [25] attaches a depth sensor to a robotic hand and plans grasps directly in the sensed voxel grid. These methods improve the 3D-models of the objects using only a single sensory modality but from multiple points in time.

Modern work in shape reconstruction has centered around pose estimation of known objects [26, 27]. The corresponding techniques utilize multi-layer-perceptrons (MLP) to map points into dense representations and perform classification tasks. They can be useful for scene segmentation, part identification, or pose estimation. Other work has divided the reconstruction and pose estimation into two separate steps to allow for a more refined mesh prediction of a set of objects on a table [28].

New work in rendering unseen images using implicit representations is catalyzing new research in the field. Sitzmann et al. [29] analyzed the effectiveness of generating meshes, images, and even sound from input data via implicit representations and novel use of sine as an activation function. Other approaches include: generating novel views of an environment from a single view [30] and utilizing text-based mesh-generation applying implicit representations [31].

Attention-based architectures for 3D shape reconstruction is an increasingly growing area of research [32, 33, 34, 35], yet most of the works focus on: (1) the bidirectional (encoder) setting to determine which regions of the input to focus on, (2) the reconstruction from the 2D images, or (3) are unable to realign the reconstruction into the real world. The attractive memorization aspect of the unidirectional (causal) attention for 3D point clouds is not well explored. Memory-based models were so far exploited mainly in the context of RNNs [36]. Multiple view networks such as [37, 38, 39, 40] are designed to reconstruct an object in a known image frame rather than completing the object in the image frame it appears in.

To leverage multiple views, an architecture that can operate on the ordered sequence of embeddings is required. There are many natural candidates for such an architecture, including LSTMs [41, 42], GRUs [43, 44], and attention-based models. MVP utilizes the last class, given unprecedented success of Transformers [11] in modeling sequential data. It uses two encoders: (1) the first one acts solely on the input frame and the other leveraging the context given by Performer’s associative memory and encapsulating all frames seen so far. As opposed to a regular Transformer, the size of this memory is independent from the number of frames seen so far. The second tower enables the input frame to attend to the compact summarization of the previous frames. (b) The outputs of both encoders are added and fed to the decoding layer.

Using Performers instead of regular Transformers is a pragmatic choice. The latter ones are characterized by linear in the number of the observed frames memory size as well as memory update (per new frame). Instead, Performers guarantee constant in the frame-history length update of the encoder towers and memory size. We explain this memory model in detail in the next section.

III. THE MULTIPLE VIEW PERFORMER (MVP) ARCHITECTURE

![MVP Encoder architecture.](image1)

(a) MVP Encoder architecture.

![MVP Decoder architecture.](image2)

(b) MVP Decoder architecture.

Fig. 3: The MVP-network takes multiple unregistered views of an object to produce a reconstruction. (a) Its encoder-part has a two-tower structure with one tower acting solely on the input frame and the other leveraging the context given by Performer’s associative memory and encapsulating all frames seen so far. As opposed to a regular Transformer, the size of this memory is independent from the number of frames seen so far. The second tower enables the input frame to attend to the compact summarization of the previous frames. (b) The outputs of both encoders are added and fed to the decoding layer.
L grows linearly in L substantial number of observations. This property is true even for the exponential-size memories. The memory (modulo attention-normalization, see Eq. 2) is given by the prefix-sum of the outer-products $\odot$ of the $\phi$-transformations of the key-vectors and value-vectors. This memory can be easily accessed and updated as new observations come. The memory effectively compresses all the recorded observations into $o_1, \ldots, o_L$ into bounded space (with size independent of the number of previous recorded observations $L$). The associative memory is updated with each new observation, without requiring a fixed window of previous observations to be tracked.

### A. MVP memory with causal Performers

All the vectors in this section are by default row-vectors. We consider a sequence of observations (image frames) $(o_1, \ldots, o_L) \in \mathbb{R}^{40 \times 40 \times 40 \times 1}$, each represented as a voxel grid with occupancy scores. In the attention approach to frame-sequence modeling, each observation $o_i$ is associated with a latent representation denoted as $v_i \in \mathbb{R}^d$ (called a value vector). Furthermore, the attention of the observation $o_i$ to the observation $o_j$ is defined as: $\text{att}_{i,j} = K(q_i, k_j)$ for two (learnable) latent encodings corresponding to $o_i$ and $o_j$, called query $(q_i \in \mathbb{R}^{d_{QK}})$ and key $(k_j \in \mathbb{R}^{d_{QK}})$ respectively and some fixed kernel $K : \mathbb{R}^{d_{QK}} \times \mathbb{R}^{d_{QK}} \to \mathbb{R}$. Attention models usually apply softmax-kernel, defined as: $K_{\text{sim}}(q_i, k_j) = \exp(q_i \cdot k_j)$. Sequence $(o_1, \ldots, o_L)$ defines the memory $M$ of the system.

For the newly coming frame $o_i$, the latent representation of the most relevant frame from the memory is retrieved approximately as a convex sum of value vectors for the frames seen so far with the renormalized attention coefficients, i.e. has the following form:

$$x_i = \sum_{j=1}^{L} \frac{K(q_i, k_j)}{\sum_{l=1}^{L} K(q_i, k_l)} v_j.$$  

(1)

This retrieval process can be thought of as a one gradient step (with learning rate $\eta = 1$) of the continuous Hopfield network with the exponential energy function [13]. If the keys of the observations are spread well enough, the procedure within a couple of gradient steps converges to the value vector corresponding to the nearest-neighbor of $o$ from $M$ (with respect to the dot-product similarity in the query-key space). This property is true even for the exponential-size memories.

This approach has a critical caveat though - the memory needs to be explicitly stored. It becomes problematic if a substantial number of observations $L$ are collected since $M$ grows linearly in $L$ and latent embedding computation from Eq. 1 clearly takes time linear in $L$.

To address this, Performers’ attention leverages unbiased estimation of the attention-kernel $K$ via linearization: $K(q_i, k_j) = \mathbb{E}(\phi(q_i) \phi(k_j)\top)$ for some, usually randomized, mapping: $\phi : \mathbb{R}^{d_{QK}} \to \mathbb{R}^m$. Such a mapping exists for the softmax-kernel in particular (see: FAVOR+/+/ mechanisms in [1, 45]). The linearization leads to the following formula for the approximation of $x_i$:

$$\hat{x}_i = \sum_{j=1}^{L} \frac{\phi(q_i) \phi(k_j)}{\sum_{l=1}^{L} \phi(q_i) \phi(k_l)\top} v_j.$$  

(2)

$$= \frac{\phi(q_i)}{\phi(q_i) \sum_{l=1}^{L} \phi(k_l)\top} v_j = \frac{\phi(q_i) M_i}{\phi(q_i) M_i,}$$  

(3)

where $M_i \overset{\text{def}}{=} \sum_{j=1}^{L} \phi(k_j)\top v_j \in \mathbb{R}^{m \times d}$ and $m_i \overset{\text{def}}{=} \sum_{l=1}^{L} \phi(k_l)\top \in \mathbb{R}^{d_{QK}}$. We call $\hat{M}_i = (M_i, m_i)$ the compact associative memory corresponding to observations $(o_1, \ldots, o_i)$. Note that the size of this memory is independent from $i$ and the update of $\hat{M}_i$ to $\hat{M}_{i+1}$ can also be done in time independent from $i$ (see Fig. 4 for the pictorial representation).

In MVP-networks, attention modules of the full Transformer-stacks used in the history-dependent encoders/towers compute latent embeddings of the frames according to Eq. 2. We apply two attention-kernels: regular softmax-kernel (with randomized mapping $\phi$ given by FAVOR+) method as well as the ReLU kernel defined as: $K(q_i, k_j) = \text{ReLU}(q_i)\text{ReLU}(k_j)\top$ for ReLU applied element-wise (and trivial corresponding deterministic mapping $\phi$). For more details see [1].

### B. The MVP Encoder and Decoder

The non-attention parts of the MVP encoder and decoder layers are inspired by the CNN architectures described by Varley et al. [8] and Yang et al. [9]. Details of the proposed MVP architecture are presented in Figure 3. The network takes $L$ unregistered views as $40^3$ voxel-grid inputs, each created by voxelizing a point cloud generated from a 2.5D depth image. All intermediate activation-functions are RELU, and the output activation-function is sigmoid (since its outputs are in the range $[0, 1]$). The intermediate representation is influenced by both the current observation and all prior observations due to the performer. The output of the Decoder is interpreted as a $40^3$ voxel-grid representing which voxels are occupied by the object independent of whether they are visible to the camera. The output voxel grid is in the same reference frame as the most recent observation.

Many alternative methods utilize a $32^3$ voxel grid resolution [46, 38, 39, 40]. We view the use of a $40^3$ voxel grid input and output as an improvement over those alternative methods. Additionally, because the input voxel data is aligned and sized with the output of the object voxel grid, the network is able to reconstruct the pose and shape of an unseen object. When evaluating the occurrence of points in each voxel of the grid, we found that on average an occupied voxel only contained 1.8 points on average. Given that objects can be placed far from a camera for
reconstruction purposes, keeping the voxel resolution at $40^3$ gives enough information for reconstruction of objects.

IV. EXPERIMENTS

A. Baselines

Three baseline methods are used to evaluate how performers impact the ability to reconstruct objects from multiple views. In the first condition, we evaluate the performance of a single-view reconstruction with a single dense layer in the embedding to evaluate a baseline floor of performance for our method. In the second condition, we modify our MVP architecture to utilize a transformer instead of a performer which we call MVT. In the third condition, we modify our MVP architecture to utilize an LSTM layer instead of a performer. In both of these cases the encoder and decoder are kept constant between Single-View, MVP, LSTM, and MVT conditions.

Additionally, we evaluate the performance benefit of training with different numbers of views. We evaluate the performance of MVP using 3, 6, and our proposed method of 12 views. We call the MVP model trained with 3 views, MVP3. We call the MVP model trained with 6 views, MVP6.

B. Experimental Setup

To evaluated performance, twenty-five total models were trained. One for each of the four model architectures (Single-View, MVP, LSTM, MVT) for each of the 5 experimental setups.

Camera Pan and Two Object Camera Pan A camera pans over the object (or pair of objects) capturing a sequence of views. The views are captured by rotating the camera around the centroid of the object.

Object Hiding The object is progressively hidden over time by a sweeping set of voxels that occlude the view. This is evaluating whether the network can continue to output the intended completion after the object has been completely occluded. This experiment is inspired by the concept of object permanence in psychology [47]. The object is hidden from view by an incremental 1 voxel thick curtain and the voxels associated with the object are removed when occluded by the curtain.

Object Reveal The object is initially hidden from view by a 1 voxel thick curtain. The object is revealed incrementally as the curtain is removed. The network will demonstrate that it can incorporate the most recent views even if no view has been provided of the object for the first few steps.

Object Slide Behind Other One object slides behind a stationary object in a the scene becoming partially or fully occluded for several views in the sequence. The start position and distance from the object closest to the camera are varied randomly.

All of these experimental conditions are shown with reconstruction examples in Figures 5, 6 and 7.

C. Training

Five different sets of data based on the YCB [10] and Grasp [48] datasets of objects are used to train the MVP model. Each of the five are based on the five experimental conditions described previously. 55 objects from the YCB dataset and 463 objects from the Grasp dataset are used. 726 views of each object are captured uniformly around each object. For the Two Object Camera Pan dataset, a random sample of the 518 objects are joined as long as they fit in a
Fig. 7: The reconstruction results from the Object Slide Behind Other experimental condition for the MVP model. The Object Slide Behind Other example shows that the MVP network will remember objects even as they are hidden behind an occluding object. All examples shown were not observed during training.

Fig. 8: Objects sitting on a table are captured and segmented from their environment. The partial view of each is then reconstructed using the MVP system by capturing multiple views of each object. The reconstructed mesh can then be used in software, such as GraspIt! [16], for grasp planning.

volume of 0.027m³ and then voxelized together. There were 925 of such pairs from the 518² possible combinations. Each of these pairs are then rendered in simulation about their centroid. For the Object Slide Behind Other dataset, 8 objects were chosen from the YCB dataset and rotated 32 different orientations for each object, capturing a total of 65536 sets of 12 views. Each of the 12 views capture the object as it slides behind the other. Each of these datasets followed a 80/10/10 split for train, validation, and test. All views in the test set are of objects not included in the train or validation sets.

Our MVP model was trained with binary cross-entropy loss and optimized using Adam. For the Camera Pan dataset, there were n = 270507 training samples of 12 views.

D. Evaluation

To evaluate the Single-View, MVP, LSTM, and MVT models, a test dataset of objects not seen during training is reserved for each of the five experimental conditions. Each view is completed and then compared against the ground truth for reconstruction quality. These views are generated by rendering in iGibson [49] and the ground truth is generated by voxelizing the mesh using binvox [50]. Following [46], we use three metrics to evaluate shape completion quality: Jaccard (MeanIoU) [51], F-score @ 1% [52], and grasp joint error [8].

We report the Jaccard similarity metric [51] as a measure of completion quality. The Jaccard similarity between sets A and B is given by:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard similarity has a minimum value of 0 where A and B have no intersection and a maximum value of 1 where A and B are identical. [8] showed that more accurate completions can be helpful for robotic grasp planning.

The F-score is a metric to evaluate the performance of 3D reconstruction results. Our implementation matches [53]. It is defined as:

$$F - Score(d) = \frac{2P(d)R(d)}{P(d) + R(d)}$$

where P(d) and R(d) are the precision and recall with a distance threshold d, respectively. The precision P(d) is defined as:

$$P(d) = \frac{1}{n_d} \sum_{g \in R} \min_{r \in G} ||g - r|| < d$$

$$R(d) = \frac{1}{n_g} \sum_{g \in G} \min_{r \in R} ||g - r|| < d$$

where R and G represent the predict and ground truth point clouds, respectively. nR and nG are the number of points in R and G, respectively. We then convert this voxel grid into a mesh by applying marching cubes to it [54] and then extract points from the surface of the mesh. We use this and the ground truth point cloud to calculate a valid F-score.

We also evaluate the grasp quality of each reconstructed object as described in [8]. We reconstruct each voxel grid as a mesh using marching cubes [54]. We then take that mesh and place it into GraspIt! [16] and perform an autograsp using a simulated BarrettHand on the mesh. We then place the ground truth mesh in place of the reconstruction and autograsp again. We then calculate the predicted versus realized grasp joint error for each joint and average over each joint. We calculate the grasp joint error for only the final predicted image in each condition. We only use this for completions of single objects, and therefore the object slide behind other and two object camera pan setups are not evaluated. Examples grasps are shown in [8]

E. Results

Sample reconstructions from MVP tests are shown in Figures [3][6][7]. In all experiments the MVP model can reconstruct images of the object well and remember features of the object that are no longer visible. For the Object Hiding and Object Slide Behind Other experiments, the MVP model was able to remember objects even as they were no longer visible. The enhancement of memory in the neural network architecture of MVP allows for a novel improvement over previous methods.

The quantitative results for the performer model are shown in Tables [1][11] and [13]. The MVP model performs better or equal to the LSTM and MVT models across all experiments in both train and test conditions. The main takeaway is that the MVP model performs similarly to the MVT and LSTM.
models. We qualitatively validated that our network can complete objects in a real world as well. To complete objects in the real world a frame is registered to segment the object from nearby surfaces. This landmark is kept track of via an Intel Realsense d415 RGBD camera. The object point cloud is captured via an Intel Realsense d415 RGBD camera. This information of the objects that are captured on a delay. We demonstrated that MVP is able to remember objects, generalize to unseen objects, and complete seen objects better than baseline methods. However, all models presented show a significant improvement over a single view. This shows that training using multiple views can improve reconstruction quality in general and is a novel methodology for training shape reconstruction networks.

V. REAL WORLD COMPLETIONS

We qualitatively validated that our network can complete objects in a real world as well. To complete objects in the real world a frame is registered to segment the object from nearby surfaces. This landmark is kept track of via an Intel Realsense T265 tracking camera. The object point cloud is captured via an Intel Realsense d415 RGBD camera. This setup is shown in Figure 2. The segmented point cloud is voxelized, then passed through the MVP system to generate an initial object hypothesis. This object hypothesis can then be used in grasp planning software like GraspIt! for robotic grasping, as shown in Figure 5. We found that for the objects we evaluated in the real world that the reconstruction quality was high for the MVP model when compared to the single-view model.

VI. CONCLUSION

This paper presented a new approach leveraging multiple unregistered views of an object to predict its mesh with higher accuracy than LSTM and transformer-based models, called Multiple View Performer (MVP). MVP critically relies on the scalable implicit-attention Transformers, called Performers, providing compact memory that can be used to utilize previous views of the scene for the shape completion. We demonstrated that MVP is able to remember objects that are no longer visible in the input and to leverage information of the objects that are captured on a delay. We also show that this shape completion system can be used for grasp planning through simulated grasping experiments. All models used in this paper are novel architectures designed to ablate the performance of the MVP. All of these models show that multiple views observed during training result in better performance for shape reconstruction in terms of reconstruction quality and grasp quality metrics.
REFERENCES

[1] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaisey, D. Belanger, L. J. Colwell, and A. Weller, “Rethinking attention with transformers,” CoRR, vol. abs/2009.14794, 2020. [Online]. Available: https://arxiv.org/abs/2009.14794

[2] M. Gualtieri and R. P. Jr., “Robotic pick-and-place with uncertain object instance segmentation and shape completion,” IEEE Robotics Autom. Lett., vol. 6, no. 2, pp. 1753–1760, 2021. [Online]. Available: https://doi.org/10.1109/LRA.2021.3060669

[3] G. Fahim, K. Amin, and S. Zarif, “Single-view 3d reconstruction: A view of deep learning methods,” Comput. Graph., vol. 94, pp. 164–190, 2021. [Online]. Available: https://doi.org/10.1016/j.cag.2020.02.004

[4] J. Moos, L. V. Gool, and M. Vergauwen, “3d reconstruction from multiple images: Part 1 - principles,” Found. Trends Comput. Graph. Vis., vol. 4, no. 4, pp. 287–404, 2009. [Online]. Available: https://doi.org/10.1561/0600000007

[5] F. Xu and K. Mueller, “Real-time 3d computed tomographic reconstruction using commodity graphics hardware,” Physics in medicine and biology, vol. 52, pp. 3405–19, 2007.

[6] A. Angelopoulou, A. Psarrou, J. G. Rodríguez, S. Orts-Escolano, F. Xu and K. Mueller, “Real-time 3d computed tomographic reconstruction using commodity graphics hardware,” Physics in medicine and biology, vol. 52, pp. 3405–19, 2007.

[7] B. Chang, S. Song, A. Verma, Y. Qi, and D. Cremers, “Photometric depth super-resolution,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 10, pp. 2453–2464, 2019.

[8] 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), 2017, pp. 2442–2447.

[9] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen, “Dense 3d object reconstruction from a single depth view,” in TPAMI, 2018.

[10] B. Calli, A. Singh, A. Wulsman, S. Sinivasa, P. Abbeel, and A. M. Dollar, “The ycb object and model set: Towards common benchmarks for manipulation research,” in Advanced Robotics (ICAR), 2015 International Conference on. IEEE, 2015, pp. 510–517.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[12] V. Likhosherstov, K. M. Choromanski, J. Q. Davis, X. Song, and A. Weller, “Sub-linear memory: How to make producers slim,” in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., pp. 6707–6719. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/55090226e4b5ec36c9ca8e4292cb0c33-Abstract.html

[13] H. Ramsauer, B. Schaff, J. Lehner, P. Seidl, M. Widrich, L. Gruber, V. Likhosherstov, K. M. Choromanski, J. Q. Davis, X. Song, and J. Varley, “Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation,” 2022. [Online]. Available: https://arxiv.org/abs/2203.01929

[14] S. Savarese, “Densefusion: 6d object pose estimation by iterative dense fusion,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[15] M. Labbé and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[16] A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011, pp. 127–136.

[17] S. Thrun and J. J. Leonard, “Simultaneous localization and mapping,” in Springer handbook of robotics. Springer, 2008, pp. 871–889.

[18] M. Krainin, P. Henry, X. Ren, and D. Fox, “Manipulator and object tracking for in-hand 3d object modeling,” The International Journal of Robotics Research, vol. 30, no. 11, pp. 1311–1327, 2011. [Online]. Available: https://doi.org/10.1177/0278364911403179

[19] M. Krainin, B. Curless, and D. Fox, “Autonomous generation of complete 3d object models using next best view manipulation planning,” in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 5031–5037.

[20] Z. Hua, J. Fu, Wang, G. Baosu, L. Yin-Sheng, and C. Yidong, “Semantic 3d reconstruction for robotic manipulators with an eye-in-hand vision system,” Applied Sciences, vol. 10, p. 1183, 02 2020.

[21] M. Krainin, B. Curless, and D. Fox, “Autonomous generation of complete 3d object models using next best view manipulation planning,” in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 5031–5037.
[37] T. Hu, Z. Han, A. Shrivastava, and M. Zwicker, “Render4completion: Synthesizing multi-view depth maps for 3d shape completion,” in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 4114–4122.

[38] A. Dai, C. Ruizhongtai Qi, and M. Nießner, “Shape completion using 3d-encoder-predictor cnns and shape synthesis,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5868–5877.

[39] 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 ECCV. Springer, 2016, pp. 628–644.

[40] K. Peng, R. Islam, J. Quarles, and K. Desai, “Tmvnet: Using transformers for multi-view voxel-based 3d reconstruction,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2022, pp. 222–230.

[41] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[42] P. Hoedt, F. Kratzert, D. Klotz, C. Halmich, M. Holzleitner, G. Nearing, S. Hochreiter, and G. Klambauer, “MC-LSTM: mass-conserving LSTM,” in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 4275–4286. [Online]. Available: http://proceedings.mlr.press/v139/hoedt21a.html

[43] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1724–1734. [Online]. Available: https://doi.org/10.3115/v1/d14-1179

[44] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Gated feedback recurrent neural networks,” in Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, ser. JMLR Workshop and Conference Proceedings, F. R. Bach and D. M. Blei, Eds., vol. 37. JMLR.org, 2015, pp. 2067–2075. [Online]. Available: http://proceedings.mlr.press/v37/chung15.html

[45] V. Likhosherstov, K. Choromanski, A. Dubey, F. Liu, T. Sarfoss, and A. Weller, “Chefs’ random tables: Non-trigonometric random features,” 2022. [Online]. Available: https://arxiv.org/abs/2205.15317

[46] M. Tatarchenko*, S. R. Richter*, R. Ranftl, Z. Li, V. Koltun, and T. Brox, “What do single-view 3d reconstruction networks learn?” in CVPR, 2019.

[47] T. Bower, “The development of object-permanence: Some studies of existence constancy,” Perception & Psychophysics, vol. 2, no. 9, pp. 411–418, 1967.

[48] D. Kappler, J. Bohg, and S. Schaal, “Leveraging big data for grasp planning,” in ICRA. IEEE, 2015, pp. 4304–4311.

[49] F. Xia, W. B. Shen, C. Li, P. Kasimbpri, M. E. Tchapmi, A. Toshev, R. Martin-Martín, and S. Savarese, “Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 713–720, 2020.

[50] P. Min, “binvox,” http://www.patrickmin.com/binvox or https://www.google.com/search?q=binvox, 2004 - 2019, accessed: 2022-05-25.

[51] S. Kosub, “A note on the triangle inequality for the jaccard distance,” Pattern Recognition Letters, vol. 120, pp. 36–38, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865518309188

[52] A. Knipitsch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” ACM Transactions on Graphics (ToG), vol. 36, no. 4, pp. 1–13, 2017.

[53] S. Yang, M. Xu, H. Xie, S. Perry, and J. Xia, “Single-view 3d object reconstruction from shape priors in memory,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3152–3161.

[54] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in ACM siggraph computer graphics, vol. 21, no. 4. ACM, 1987, pp. 163–169.