3D Object Representations: Analysis, Suitability and Exploration of NeRFs vs Point Clouds

Ananya Sane*
IIIT Hyderabad
ananya.sane@students.iiit.ac.in

Gaurav Singh*
IIIT Hyderabad
gaurav.si@research.iiit.ac.in

Yash Mehan*
IIIT Hyderabad
yash.m@research.iiit.ac.in

***Abstract*—There are multitude ways to represent a 3D object, each of which is optimised for and serves a specific purpose. Grasping is a fundamental yet nuanced and complex skill that is hard to perfect. It demands that robotic perception, planning and control work in tandem. In cluttered, unstructured environments, grasping is still far from human-like accuracy. We explore how using different representations affect the grasp and performance. We run a state-of-the-art grasping method - Contact GraspNet on different representations to compare, contrast and analyze why certain representations perform better than others. Then, a discussion on the recent data-driven techniques in grasping that have developed in recent years follows. With the advances in computer vision, we see investigations into semantic grasping, a technique that could, in the future, be the basis of intelligent manipulation. Finally, we discuss the current problems and further scope for research.**

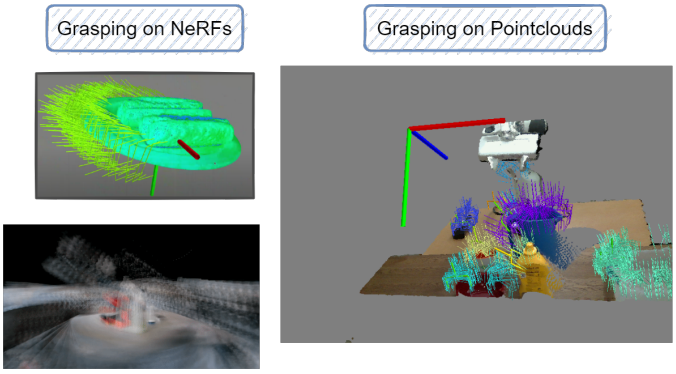

Fig. 1. Grasping on Different 3D representations

I. Introduction

There exist a multiple ways to represent 3D objects for robotic perception, namely, point clouds, depth maps, voxels, SDFs, NeRFs etc. Each of the representation is suited for a niche purpose. In this article we aim to analyse the performance and suitability of different representations, in context to grasping 3D objects. For humans grasping objects, novel or otherwise, is trivial. For robots to accomplish movement of any object in its environment, it needs to be able to determine the optimal grasp: not only is ascertaining where to place the grasp a challenge, but also which among the prospective grasps, predicting the probability that a given grasp will successfully hold the object. A critical requirement for grasping is effective registration and perception of the subject. Developing robots which estimate gripping in dynamic and unstructured environments is still a challenging problem today. An evaluation of grasping on different representations namely NeRF and depthmaps will be carried out and analyzed.

Point clouds represent an object or a scene as a collection unstructured points, in the explicit euclidean space. The representation is an unordered collection of points and has no information pertaining to the correlation among the set of points. The major advantage of this representation i s the flexibility that comes along with it. It is trivial to register and project points from multiple sensors at various locations in space and in time, so long as the relative or absolute transform among them is known. Other advantages are that point clouds can have variable density and the user is not bound to the world size for the representation, as the representation is essentially only an unordered collection of coordinates. Such metadata about the scene or the objects under examination need not be known apriori. Like point clouds are quite a suitable choice for extensible spaces and registration, there are significant caveats that follow. Point clouds suffer from their lack of inter-point structure, lack of information as to which how a surface or volume spans as there is no way to express any two points in a point cloud are physically connected. This implies that point clouds must be sampled very densely with respect to occupied regions and surfaces, lest ambiguities can still arise. There is scope for ambiguities when an object has small orifices, or very detailed surface or when multiple thin surfaces run parallely. A drawback of point clouds in specific context for grasping is it limited by the resolution and constraints of the camera sensor equipment; since it captures discrete points, it might not capture the surface in details. The grasp prediction model thus has to interpolate between the points to create a surface to grasp on, which might not correspond to an appropriate grasp for objects in the real world. Although point clouds are not computationally expensive to transform in the representation space, however, resizing point clouds can require point interpolation to preserve density. Voxel representations can be converted to point clouds by sampling at the center of every voxel. Converting point clouds to meshes involves Delauney Triangulation which determines the optimum configuration of assigning points to elemental triangles. For converting to voxels, Cube marching algorithm

is employed.

Depth-images are essentially images which capture depth information of a scene in place of RGB pixel information. The depth maps are said to be two-and-a-half-D structure of the scene [1]. Given a depth image, (generally from an RGB-D camera or a LiDAR sensor), and the camera intrinsics, we can generate the point clouds corresponding to each pixel in the depth map. Due to this, depth images and point clouds are can be considered interchangeable given little information about the intrinsics. Most pipelines convert depth maps to point clouds as a pre-processing step, and operate on the point clouds. Stitching multiple such depth images culminates in a formation of a depth map. The image representation of the depth map may not lend itself nicely to standard image compression techniques, which are qualitatively evaluated by human eyes. Feature extraction on depth maps by CNNs is very well researched field. Feature extraction is in essence the first step for an intelligent system to ascertain an optimum and a firm antipodal grasp, for instance, DexNet [2] for grasp proposal on depth images and GQCNN [3] for evaluating how good a grasp is. The grasp is evaluated in terms of the predicted probability that the grasp will be able to hold the object while lifting, transporting, and shaking the object.

Voxel based representations are the most natural extension to 2D data representations to 3D and model a 3D object as a volumetric image, spanning a 3D euclidean space. The voxel, at a location in the grid, may store a vector if multidimensional, a value, or may have just a binary value indicating occupation or vacancy. The commonest approach to implementing voxels is an occupancy grid. The advantage to this is that common retrieval operations, collision and obstacle detection are constant time complexity. Having a grid, however, implies challenges on extensibility during runtime, and the world size is fixed, or consumes significant compute and storage resources to reallocate memory for increasing world size. Another limitation is that rotations of objects in increments of angles not a multiple of 90 degrees is lossy and requires interpolation and leads to loss of voxels due to discretization of the euclidean space into rigid cubes. As a result, repeated rotations will lead to loss of detailing and degraded structure. Since voxels are dense, it is possible to express interiors of hollow objects. However, present approaches can only handle modest resolutions due of the cubical increasing compute and memory needs. As a result, voxel-based techniques are unsuitable to represent minute or highly detailed shape features. In addition, because voxels' normals are not smooth when rendered, they visually differ significantly from high-fidelity shapes.

Mesh based representations encode objects using data structures that enables the depiction of a set of connected polygonal subdivisions of a surface, usually implemented as a graph where the nodes are the points on the surface and the vertices of the polygons and the edges are the edges of the graph. Meshes are meant to represent Meshes are frequently used in modelling to discretize a continuous or implicit surface as well as in computer graphics to depict surfaces. A mesh is comprised of edges connecting vertices (or vertex), which form the faces (or facets) of a polygonal shape. We refer to triangular meshing when all faces are triangles. Meshes are lightweight and optimised for representing surfaces, and not volumetric density, and include the caveat of not being able to distinguish between hollow and occupied 3D objects.

Recently, implicit representations parameterised by neural networks, often known as Implicit Neural Representations or INRs, have been gaining popularity. Out of them, Neural Radiance Fields (NeRFs), have gained immense popularity in their ability to represent radiant and volumetric density from a set of multi view images. Within 2 years, the training time for NeRFs has decreased to under 2 minutes, because of which many tasks that were earlier done on explicit representations such as pointclouds, are now being studied for implicit representations.

II. Related Work

A. Robotic Grasping

Data-driven strategies are currently being used to overcome the robot grasping challenge. Contrary to earlier approaches, which operated on manually created feature vectors, more modern methods, leverage convolutional architectures to operate on unprocessed visual measurements. By portraying the grasp as an orientated rectangle in the image, the majority of these grasp synthesis techniques are made possible. The gripper posture must be parallel to the image plane in order to comply with this 3-DOF representation. The disadvantages of such a depiction are numerous: Since it restricts the variety of grasps, picking up an object can be impossible due to extra task- or arm-related restrictions. It also results in a very constrained workspace in the case of a static image sensor.

B. Transparent and Specular Object Grasping

Grasping transparent and specular objects is quite a challenging problem. Most depth-based methods fail due to incomplete depth maps, due to the subject being transparent or specular, so are not registered well by ordinary depth sensors usually, because of the reflection and refraction of light from the subjects. One way is to restore the lost depth information before inferring the grasp. ClearGrasp [6] is one such method which firstly estimates shape information from a single given image, initialises a depth map and then optimises for the depth. [7] is one such dataset of depth images of real world objects. [8] is a grasp predicting model which learns from an existing depth-based model. Other advances include Swin Transformer-based RGBD fusion network, SwinDRNet, for depth ascertainment and 6DoF grasp calculation. Another exploratory field has been predicting grasp poses Other studies focus on predicting grasp poses given only RGB and depth. Other approaches have been predicting grasp poses from the RGBD-based CNN. Single-view-based grasping methods do not handle occlusion cases well. GlasLoc contructs a Depth Likelihood Volume descriptor from the given multiple RGBD images.

C. Neural Radiance Fields

Neural Radiance Fields (NeRFs) are implicit volumetric representations of 3D scenes that encode their look and shape. A continuous representation of a 3D scene is stored by a NeRF model ϕ within the weights θ. It can be seen as a function that converts a point in the scene, x, and the direction of seeing, d, to a view-dependent brightness, c, and a view-independent density τ

$$c, \tau = \phi(x, d; \theta) \tag{1}$$

Given a ray specified by the camera pose d and origin o as

$$r(\lambda) = o + \lambda d \tag{2}$$

, the volumetric rendering equation is defined as follows:

$$C(r) = \int_0^{\lambda} \mathcal{T}(\lambda; r).\tau(r(\lambda)).c(r(\lambda), d)d\lambda \tag{3}$$

$$\mathcal{T}(\lambda; r) = \exp\left(-\int_0^{\lambda} \tau(r(u))du\right) \tag{4}$$

where $\mathcal{T}(\lambda; r)$ refers to the transmittance or the transparency of a particular point, which is in turn defined by the volumetric density $\tau(r)$. This parameter defines the probability by which a ray intersecting with the point passes through it. The network parameters θ are optimised over an l2 loss on the color obtained through the volumetric rendering $C(r)$, from the color of the pixel in the i^{th} original image C^i_{gt} :

$$\mathcal{L} = \sum_i \mathbb{E}_{r \sim I_i}\left[||C(r) - C^i_{gt}(r)||^2_2\right] \tag{5}$$

It is important to notice how transparency as well as volumetric density are related in a NeRF. This might hint to one flaw that transparent objects are considered less dense by a NeRF model, (which, althought theoretically may be more sound, practically is not the case as shown by Dex-NeRF [18])

D. Implicit Representations for Grasping

Recent work in implicit 3D data representations has brought about research in their usage in the realm of robotic grasping by leveraging implicit scene representations for robotic tasks. GIGA seeks to use the relationship between geometric reconstruction and grasping prediction, using the convolutioonal occupancy network to construct the implicit representation. Due to the fact that the input is a single view depth image, it is difficult to reconstruct transparent objects. NeRFs are used to create the depth maps and provide correspondence for desciptors of objects in the manipulation task. DexNeRF is the first to use NeRF in the context of grasping transparent objects throught transparency aware depth rendering. However, this comes at the cost of hours of training the NeRF per grasp, ehich is not practical for real world, non structured environments. It is a 3 DoF grasping, which poses some limitations. EvoNeRF speeds up vanilla NeRF optimisation through the usage of Instant-NGP, but still cannot avoid dense inputs and training per grasp. GraspNeRF is free from these caveats of scenewise optimisation, requirement of dense input, and is thus suited to real world grasping tasks.

III. Background

Grasp Proposal network: We chose Contact GraspNet [12] as our grasp proposal network, which is a SOTA 6-DoF grasp proposal network for pointclouds. Existing pipelines in the domain of robotic grasping with full 6 DoF are complex and posses several failure points. Contact GraspNet is able to compute parallel-jaw grasps directly from depth recordings. It treats the 3D points of the pointcloud as prospective grasp points. Contact GraspNet specialises in generating grasps on objects in cluttered scenes, and the model the researchers trained has achieved over 90% success rate on real world cluttered scenes. Grasping based on models of known objects removes the need for reasoning the physics of the grasp generation and contact as it defines grasps in the object's frame, which are then transformed based on the 6 DoF detected keypoints or object poses. However, the limitations of such an approach is that it can only work on a limited set of objects, and will be error prone otherwise. Thus, Contact GraspNet takes the Model-free approach, where there are no definite assumptions made regarding the shape or type of the object, and there exists a common representation of variously shaped and sized objects. But in addition to the large SE(3) space for grasps, the problem of learning becomes quite challenging. Grasping in a cluttered scene is an interesting problem as not only do grasps for an object need to be successful, they must also not collide with other objects in the scene, while still taking into account the motion constraints of the robot that is performing the grasping itself. Contact GraspNet uses partial pointclouds to generate grasps, with optional segment maps with object masks to distinguish different objects that are then used to eliminate grasps that cause collisions. An overview of the working of Contact GraspNet is as follows. It is assumed that a minimum of one of the two grasp points are visible before grasping, as grasps with no visible grasp points are often ambiguous. The grasp pose g is estimated as below:

$$t_g = c + \frac{w}{2}b + da$$

Here, a is a unit approach vector, b is the unit grasp baseline vector, d is the constant distance from the gripper baseline to the gripper base, c is the contact point in the 3D point cloud and w is is the grasp width.

Contact GraspNet uses PointNet++ to process the pointclouds and extract features in local regions of the scene.

Since our experimentation involved grasping on pointcloud scenes obtained from other representations in various methods, we chose Contact GraspNet to gauge the efficacy of grasp generation on the above mentioned objects.

Structure-from-Motion SfM is a technique to reonctruct the 3D structire of a scene or an object,by projecting it onto a series of photos taken from various angles. A sequential processing pipeline called incremental SfM (referred to as SfM in this study) has an iterative reconstruction component. Typically, feature extraction and matching are done first, then geometric verification. The generated scene graph serves as the basis for the reconstruction step, which incrementally registers

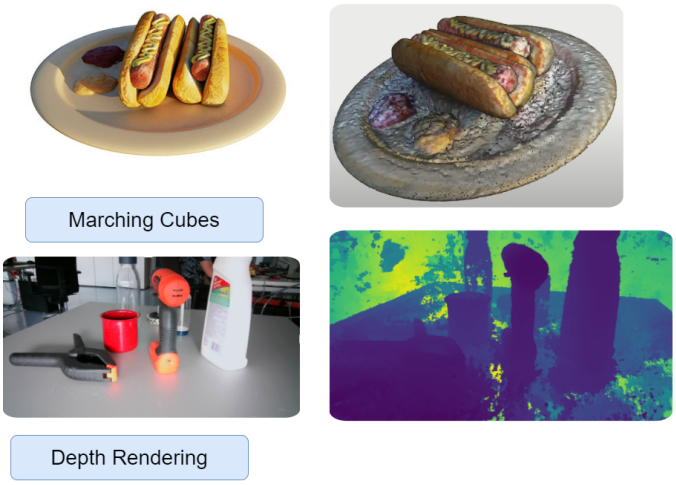

Fig. 2. Conversion to pointclouds, the two methods for converting to pointclouds are shown. **top**: depth map pipeline. **bottom**: marching cubes pipeline

additional images, triangulates scene points, filters outliers, and fine-tunes the reconstruction using bundle adjustment after starting the model with a carefully chosen two-view reconstruction (BA).

Dex-NeRF: Dex-NeRF [18] targeted a niche area where NeRFs could arguably perform much better than pointclouds, which was the grasping of transparent objects. The assumption made was that most available pointclouds come from depth maps that are taken from either a LiDAR or a depth camera, both of which (especially a depth camera) give bad depth estimation and subsequent pointcloud construction when transparent objects are present in a scene. NeRFs, however, also capture the radiance and thus are able to take into account the reflections/refractions of the transparent objects as an alias to the emitted light from a surface and thus by design assign some volumetric density, and transitively geometric structure to transparent objects. The structure-from-motion pipeline is run as a preprocessing step before a NeRF is trained, as NeRF requires the camera poses to be there. The state-of-the-art SfM pipeline COLMAP allows NeRFs to train on videos or images taken from a smartphone without any calibration. We employ the COLMAP pipeline in our experiments and also show the limitations of such a pipeline.

Instant NGP: To try to reduce the training time for NeRFs, which is one of the major issues currently, Nvidia's Instant NGP [4] tries to lower this cost by introducing an adaptable new input encoding that enables the use of a smaller network without sacrificing quality, thereby drastically lowering the number of floating point and memory access operations: a small neural network is supplemented by a multiresolution hash table of trainable feature vectors whose values are optimised through stochastic gradient descent. By employing fully-fused CUDA kernels to implement the entire system, which allows for simple parallelism while minimising wasted bandwidth and compute operations, the network is able to distinguish between hash conflicts thanks to the multiresolution structure.This obtains a combined speedup of many orders of magnitude, making it possible to render at 1920 x 1080 resolution in tens of milliseconds and train high-quality neural graphics primitives in just a few seconds.

IV. Experiments

A. Experimentation methodology

To study the difference in grasp performance and the changes in the fundamental formulation of the problem of robotic grasping with changing representations, we propose multiple experimental pipelines. Each pipeline is built with the same grasp proposal network [12] at the end to avoid introduction of other unalterable variables which might creep in due to different grasping models, which affect the depth to which the nature of the 3D representations can be analysed and jeopardizing the aim.

The following pipelines were assembled

- Grasping directly on point clouds
- Grasping on depth maps derived from a NeRF
- Grasping on point clouds derived from a NeRF via ray marching.

These pipeline were run on curated data as well self collect data. One of the objectives of this study is to demonstrate that for explicit representation, given incomplete information, extrapolation and inference are often noisy and inconsistent. Whereas Implicit representations can be sampled from, with arbitrary precision and have much wider applications because they incorporate radiance.

B. Grasping on Registered Pointclouds

Contact Graspnet shines in the domain of grasping in structured cluttered environments. It takes pointclouds, and optionally depth and seg maps as inputs to generate grasps for each object present in the scene.

Datasets Used: For grasping in registered pointclouds, the pointclouds used to test the model from the paper were used, along with their associated segmaps and depth information. These pointclouds were of cluttered table-top scenes taken from various angles.

Experimentation: Initial experimentation involved running the trained model directly on the given dataset to observe the grasps obtained. In further experimentation, the given pointclouds were downsampled to 1%, and the model was run on on these new pointclouds in order to observe the effect of low point density on the efficacy and accuracy of the grasps. Pointclouds of scenes containing transparent objects (like glassware) were also used to test the model.

C. Grasping on NeRFs

Grasping directly on NeRFs is tricky because NeRF is an implicit representation, which implies that it can't be directly operated on as easily as point clouds. In order to infer geometry or topology from surfaces, one has to either employ methods that do so in the implicit space itself, for example: operators (such as partial differentials or Fourier transform)

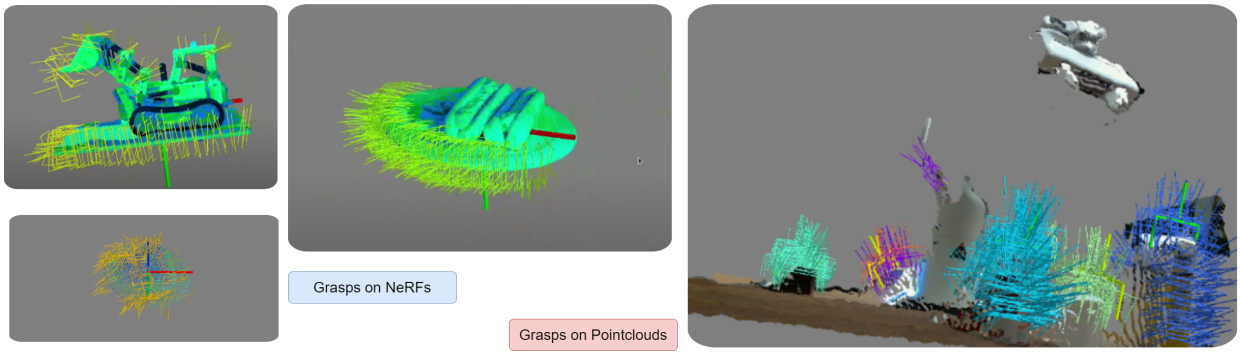

Fig. 3. Qualitative results of grasping on different representations

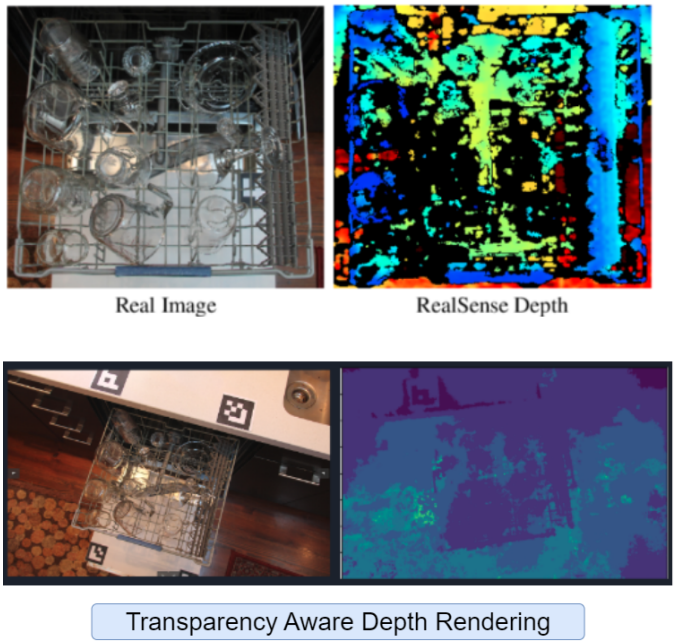

Fig. 4. Results showing the effect of transparency aware depth rendering when compared with a depth camera.

that operate on the continuous function of NeRF to give a function parametrizing the surface, or the change in colour. Such methods are beyond the scope of this study. Here we look at grasping on NeRFs by converting it into 2 explicit representations namely, depth maps, and point clouds. One must ask why a depthmap conversion is even necessary as our grasp proposal network extrapolates depth maps to point clouds in order to generate 6-DoF grasps; and given that depth maps are only from one view, their point clouds are always partial, unless registered. Our NeRF grasping pipelines explain the reasons behind it.

Datasets Used: For grasping in NeRFs, we required NeRF datasets that were of small table-top scenes, or of objects that are graspable, rather than larger scenes. We used a combination of the original NeRF dataset, namely the lego and hotdog data images, combined with with datasets released with [18]. We also tested the NeRF reconstruction on our own dataset, using COLMAP as a preprocessing step, analysing the performance of grasping in NeRF in novel scenes with uncalibrated cameras and noisy SfM output.

Conversion to pointclouds: In order for contact graspnet to work, we can to perform an implicit to explicit conversion on NeRFs to pointclouds. We did this via two methods:

1. NeRF $\longrightarrow$ Depth Map $\longrightarrow$ Pointcloud
2. NeRF $\longrightarrow^{marchingcubes}$ Mesh $\longrightarrow$ Pointcloud

The first pipeline involves a transparency aware depth rendering to a depth map and the converts the depth map to pointclouds, since the camera intrinsics and extrinsics are known. Although this gives an incomplete pointcloud, that is only from one angle, this is able to better render transparent objects and thus giving a clearer pointcloud as compared to direct ray marching. However, because of the information loss in the depth map, the pointclouds generated are usually sparser in this pipeline.

The second pipeline involves using the marching cube algorithm to construct a triangle mesh out of the NeRF which is then uniformly sampled across all surfaces at variable densities to get the pointcloud of the scene. This allows us to convert the entire NeRF to a pointcloud in one go without requiring to register multiple pointclouds generated from NeRFs, capturing occluded areas better and allowing for sampling to be done at any density.

V. Results and Analysis

A. Grasping on Pointclouds

On registered pointclouds, the results were identical to those of the paper in the case where the pointclouds used were the same. When the downsampled pointclouds were used (Fig. 5) to generate grasps, however, we found that at lower densities, the grasps were no longer accurate or physically feasible. Due to the sparsity of the pointclouds, it was clear that the grasps generated were going into the object itself. Thus, it was found that the model is robust to lower pointcloud densities, but only up to a threshold, beyond which the quality of the grasps quickly deteriorates.

In the case of pointclouds of transparent objects (Fig. 4), it was found that the grasps were inaccurate regardless of pointcloud density. This is due to the fact hat pointclouds

Fig. 5. Results showing the difference in grasping directly on dense and sparse pointclouds. We can clearly see that dense pointclouds give much better grasps while the grasps are colliding with the pointcloud in the case of sparse pointclouds as the network isn't able to distinguish gaps with the low density of the pointcloud. In an explicit representation, there is no information that we can use to make the pointclouds dense again, whereas NeRFs can be sampled at arbitrary densities due to their implicit nature.

Fig. 6. Results showing noise generated during the training of NeRF. As the background objects are usually visible at low sharpness and are often occluded by the main objects at focus, they end up acting as noise in the training process. This is a prohblem for grasping as we need to add another processing step after training of the nerf that segments out the objects of interest.

inherently fail to capture the structure of transparent objects accurately. Pointclouds rely on RGB data which is not easily obtained in the case od transparent objects. Thus the model did not accurate grasps.

As a whole, pointclouds are a convenient 3D representation for the problem of grasping, provided that they are sufficiently dense. They are platform independent and extremely versatile. They also have smaller memory and computational requirements as they are essentially just a set of coordinates. But there do exist drawbacks to using them, as seen in this study. Another drawback is the fact that one cannot have arbitrary precision in terms of sampling on pointclouds, unlike NeRFs. One is forced to use only the existing points and any sort of extrapolation (surface and normal computaion, for example) is computationally complex.

B. Grasping on NeRFs

A dominating advantage of using implicit representations like NeRFs is the capturing of transparent objects which otherwise are not captured well. Transparent objects like glass etc. do not show up in depth maps because for the depth sensors (quite often LiDAR) such objects do not reflect back.

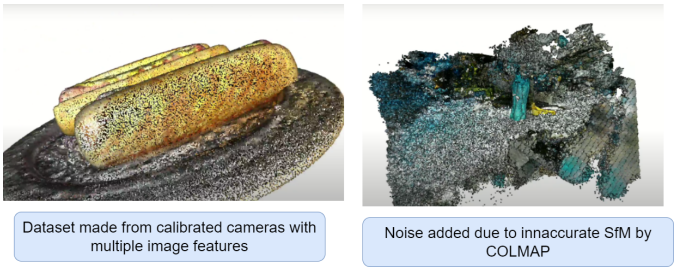

Fig. 7. Results showing noise generated due to the inaccurate structure-from-motion by COLMAP. This is usually due to less number of extracted features which results in noisy correspondences and trassitively, a noise NeRF

This is because NeRF involve the use of radiance which is derived from reflections and refractions from the objects captured.

Generating a NeRF from self procured data generates noisy outputs using COLMAP, the state-of-the-art SfM pipeline. Since grasping methods currently need an explicit conversion from the implicit ones, there is even more scope for the representations to incorporate artefacts and noise. Hence SfM and explicit conversions are the bottleneck to this procedure. There is a need for calibration marks for training NeRFs as well. The noise in self procured data was sufficient for the pipelines to produce erratic and faulty results. This implies data curation step is critical and significant care must be taken at all steps to denoise the data. Although there is extensive ongoing research in these fields and these limitations may soon be solved.

VI. Discussion

We see from the above results that NeRF is a much suitable representation for grasping overall if the sources of noise can be reduced and the marching cube algorithm be made more accurate. Looking at the above inferences we propose a **hypothetical novel combined representation**, that combined the advantage of pointclouds in localizing the geometric information to the objects that are in focus and important, with the implicit nature of NeRFs and its ability to store radiance as well as the benefit of sampling at arbitrary densities. We thus propose a combined representation that is defined on N points $X = \{x_1, x_2, \ldots, x_n\}$ where the radiance and volumetric density is defined on each point as the center of a implicit gaussian function defined again on radiance and volumetric density. During training, the Neural network first optimises the location of the points X and then the standard deviations $\{\sigma_{c(r,d)}, \sigma\tau(r,d)\}$ for each point. This allows this representation to take advantage of both implicit nature and to overcome the noise issues with background artifacts and be computationally efficient during training as the raymarching and volumetric rendering steps, which are the most computaionally heavy steps, are localised to specific point locations rather than over the entire 3D space.

VII. Conclusion

Recent breakthoughs in this field have provided better implicit representations and implicit-explicit space transformation techniques, and models to predict grasps on such representations. We observe how implicit representations are able to capture more and are independent of sample constraints, but need external calibration, compute support and denoising interventions. We observe how implicit representations are interchangeable and suffer from interpolation ambiguities but current capturing methods fail to capture details which affect tasks like grasping later in the pipeline. Our observations suggest there are. And thus we propose a representation to help mitigate such issues. We foresee a lot of lacunae of robustness with respect to noise, sampling density, and compute constraints to solve in this field.

References

[1] https://cvit.iiit.ac.in/research/projects/cvit-projects/depth-image-representations

[2] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, Ken Goldberg, "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics"

[3] Maciej Jaśkowski (1), Jakub Świątkowski (1), Michał Zajac (1), Maciej Klimek (1), Jarek Potiuk (1), Piotr Rybicki (1), Piotr Polatowski (1), Przemysław Walczyk (1), Kacper Nowicki (1), Marek Cygan, "Improved GQ-CNN: Deep Learning Model for Planning Robust Grasps"

[4] Thomas Müller, Alex Evans, Christoph Schied, Alexander Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding"

[5] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis"

[6] Sajjan, Shreeyak S. and Moore, Matthew and Pan, Mike and Nagaraja, Ganesh and Lee, Johnny and Zeng, Andy and Song, Shuran, "ClearGrasp: 3D Shape Estimation of Transparent Objects for Manipulation"

[7] Hongjie Fang, Hao-Shu Fang, Sheng Xu, Cewu Lu, "TransCG: A Large-Scale Real-World Dataset for Transparent Object Depth Completion and a Grasping Baseline"

[8] T. Weng, A. Pallankize, Y. Tang, O. Kroemer, and D. Held, "Multi-modal transfer learning for grasping transparent and specular objects," IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3791–3798, 2020.

[9] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large scale benchmark for general object grasping," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11 444–11 453.

[10] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in Conference on Robot Learning. PMLR, 2021, pp. 1602–1611.

[11] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," Robotics: science and systems, 2021.

[12] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact graspnet: Efficient 6-dof grasp generation in cluttered scenes," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.

[13] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgbd images," in Proceedings of the International Conference on Robotics and Automation (ICRA), 2021.

[14] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Grasp ness discovery in clutters for fast and accurate grasp detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 15 964–15 973.

[15] H. Fang, H.-S. Fang, S. Xu, and C. Lu, "Transcg: A large-scale real world dataset for transparent object depth completion and a grasping baseline," IEEE Robotics and Automation Letters, pp. 1–8, 2022.

[16] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 3634–3642.

[17] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang, "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in European Conference on Computer Vision (ECCV), 2022.

[18] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," in Conference on Robot Learning. PMLR, 2022, pp. 526–536.