A Review of Scene Representations for Robot Manipulators

Carter Sifferman SIFFERMAN@WISC.EDU
Department of Computer Sciences
University of Wisconsin - Madison

1. Introduction

For a robot to act intelligently, it needs to sense the world around it. Increasingly, robots build an internal representation of the world from sensor readings. This representation can then be used to inform downstream tasks, such as manipulation, collision avoidance, or human interaction. In practice, scene representations vary widely depending on the *type of robot*, the *sensing modality*, and the *task* that the robot is designed to do. This review provides an overview of the scene representations used for robot manipulators (robot arms). We focus primarily on representations which are built from real world sensing and are used to inform some downstream robotics task.

Building an intermediate scene representation is not necessary for a robotics system. It is completely possible for a robotics system to act directly on sensor data (e.g. predict appropriate grasps directly from RGB images), and we will look at many such systems within this review. However, we argue that intermediate scene representations are beneficial for robot manipulators as they:

- act as **spatial memory**
- are **efficient storage** of past memories
- allow **long-horizon planning**
- can act as **regularization** and encode **spatial priors** for learning systems

In this review, we organize scene representations into three categories depending on the task that the representation supports. These categories make up the sections of our review: collision avoidance (section 2), manipulation (section 3), and teleoperation (section 4). Within each section, we provide a review of the existing literature, summarize the challenges in the area, and consider directions for future research. In section 5, we look at scene representations for manipulators as a whole, and consider directions for future research which cut across our three categories.

1.1 Literature Survey Process

Our literature survey consisted of two phases. In phase 1, we performed a broad search of existing reviews in order to gain context and understand the broader landscape of robotics.

Figure 1: This review focuses on robotics applications which build some intermediate scene representation between sensor data and a downstream robotics task. Image sources: [1, 2, 3]

Category	**Snowball Seeds**	**# Papers Found**
Collision avoidance	[2, 18, 3]	26
Manipulation	[19, 20, 21]	28
Teleoperation	[22, 23, 14]	16

Table 1: The three sections that this review is organized into, and the "snowball seeds" which began the literature review process.

To find these reviews, we use Google Scholar search with the "Review Articles" filter enabled. The result of this search is 13 reviews spanning a broad spectrum: manipulation and grasping [4, 5, 6, 7, 8, 9, 10], SLAM [11], human-robot interaction [12, 13], teleoperation [14], motion planning [15, 16] and inverse kinematics [17]. Knowledge gained from these reviews was used to determine our taxonomy of scene representations, and the scope of this review.

The goal of phase 2 of our survey process was to find directly relevant papers which employ a scene representation for robot manipulators. As the papers that we search span a wide range of topics, keyword searches did not prove effective. Instead, we "snowballed" through references, beginning at seed papers which were found through keyword search, reviews, or consulting with colleagues. These seeds are shown in table 1. We snowballed through references both through traditional reference chasing, and using Google Scholar's "Cited By" page to find papers published after the seed paper. We also use the Abstract Viewer Project[1], a system for finding related papers developed as a research project at UW-Madison. Abstract Viewer does not use citations to find related papers, instead relying on analysis of textual content. This proved helpful for finding still-related papers when references ran dry. In total, 69 papers were collected in phase 2.

This review is not meant to act as a comprehensive survey of any one subject. Instead, we hope to give a broad overview of the field and provide jumping-off points for further reading. Articles were not selected to be a representative sample of the entire field; papers which use an identical scene representation to existing work may be excluded while those with a unique scene representation are generally included.

1. https://pages.graphics.cs.wisc.edu/AbstractsViewer/

2. Collision Avoidance

A central challenge in robotics is being able to avoid collisions, which can potentially be very costly with a powerful, fast, and expensive robot. Some approaches for collision avoidance respond directly to measurements from robot-mounted sensors, without building any intermediate representation [24, 25, 26]. A downside of this approach is that it has no memory: the robot can only act based on what it currently observes, and cannot plan based on its previous observations. As a result, these approaches are overly cautious and perform poorly in challenging conditions.

Modern approaches for collision avoidance can be broadly grouped into two categories: motion planning [27], in which the start and goal position of the robot end effector are known, and the goal is to find a viable, collision-free path between the two, and inverse kinematics, in which the end effector position is known, and the goal is to find a viable joint configuration of the robot which matches that end effector position. This distinction is unimportant for this review, as both of these approaches have the same requirements of their scene representations: fast collision checks and (sometimes) fast calculation of the distance to the nearest obstacle or direction to the nearest obstacle. As a result, scene representations for collision avoidance are similar whether the problem is formulated as inverse kinematics or motion planning.

Potential Fields– Early approaches to collision avoidance in a motion planning context represented the environment with a potential field, first proposed in [28]. This potential field can be evaluated at any point to yield a scalar "potential" value. This value is determined by both the scene geometry (which have high potential around them) and the desired location (which has a low potential). In order to move through space, the robot simply follows the negative gradient of this potential field. The potential field was heavily utilized in early motion planning work [29, 30, 31]. While effective for the time, potential fields suffer from a few problems: the potential function is prone to local minima, and can be very difficult and computationally intensive to construct [15]. Additionally, it is impractical to construct potential fields in real-time from sensor measurements, both because they are slow to construct, and because they require scene geometry to be described in a friendly closed-form, which sensors cannot provide natively. Later work [32] improved on the local minimum problem by taking into account the relative starting position as well as scene geometry during potential field construction, but nonetheless potential fields have fallen out of favor in collision avoidance applications since the early 2000s.

Signed Distance Fields– A signed distance field is a mapping between 3D points in space $\mathbf{x}$ and the scalar distance d to the nearest obstacle:

$$SDF(\mathbf{x}) = d$$

The SDF has the nice property that taking $-\nabla SDF(\mathbf{x})$ yields a vector pointing towards the nearest obstacle. This representation has been used in computer graphics since at least 1998 [33, 34], and became popular for robot collision avoidance with the introduction of the highly influential CHOMP motion planner in 2009 [3]. CHOMP uses the signed distance field, along with pre-computed gradients to perform optimization over the robot configuration. Subsequently, the popular STOMP [35], TrajOpt [36], and ITOMP [37] motion planners also used a signed distance field to represent their environment, and used gradients in a

similar way. In each of these works, the signed distance field is computed at fixed points on a regular voxel grid as a pre-processing step. To do such a computation, a precise model of the underlying geometry is needed, typically in the form of a mesh. Similarly to potential fields, computing a signed distance field is computationally expensive, and generating it from noisy sensor data is difficult. In practice, collision avoidance approaches which use an SDF are constrained to simulations, where the SDF can be pre-computed, or static environments in the real world. Regardless, SDFs are the most popular scene representation for collision avoidance, largely because they are supported by popular and effective motion planners. There exist libraries such as VoxBlox [38] and FIESTA [39] which efficiently compute and store discretely sampled SDFs for this purpose.

Collections of Primitives– A less common method for representing geometry for collision avoidance is with a collection of primitive shapes, such as spheres, cylinders, and cubes. These primitives are usually stored parametrically, so that collision checking can be done quickly and the objects represented natively in optimization solvers. Toussaint et. al. [40] proposed Logic-Geometric Programming, in which the scene is composed entirely of parametric cylinders, blocks, and planes. This paper has been influential for its elegant optimization-centered formulation, but the scene representation used within has not been heavily utilized; it serves more as a demonstration of the approach's capability. Similarly, Gaertner et. al. [41] considers collision avoidance with humanoid robots, and uses a collection of primitives to represent dynamic scenes. Similarly to Toussaint, the collection of primitives is used primarily to demonstrate the capabilities of the system. In contrast, Zimmerman et. al. [42] proposes a method for using collections of primitives in gradient-based optimization methods such as TrajOpt [36]. They provide a unified method for dealing with many types of primitives, and a method for taking the derivative of the distance to the nearest primitive, making a collection of primitives a drop-in replacement for SDFs. However, this approach has not seen widespread adoption.

Collections of Convex Hulls– A collection of convex hulls is another less common way to represent scene geometry for collision avoidance. Similarly to primitive shapes, convex hulls allow for fast collision checking and natural representation in optimization solvers. Convex hulls have the additional benefit that any shape can be broken down to a collection of convex hulls via an algorithm like QuickHull [43]. Schulman et. al. [44] uses a set of convex hulls to represent a scene, and outperforms the SDF-based motion planners of the time like CHOMP [3] and STOMP [35]. CollisionIK [18] introduces an optimization-based method for *inverse kinematics*, which is able to operate in real-time (e.g. for mimicry control) and avoid collisions with dynamic obstacles. CollisionIK mentions that point cloud objects could be broken down into convex hulls in real time, but does not demonstrate such a process. To our knowledge, no existing approach constructs convex hulls in real time from sensor data.

Learned Representations– Within the last year, one approach has emerged which uses a learned environment representation to enable real-time collision avoidance with dynamic obstacles and real-world sensing. This paper is somewhat influenced by the growing literature around learned representations in computer vision [45], graphics [46], and SLAM [47, 48]. RCIK [2] proposes a collision cost prediction network, a neural network which takes as input features extracted from an occupancy grid, as well as a 3D point in that grid; from this input the network predicts the collision cost, which is an approximation of the

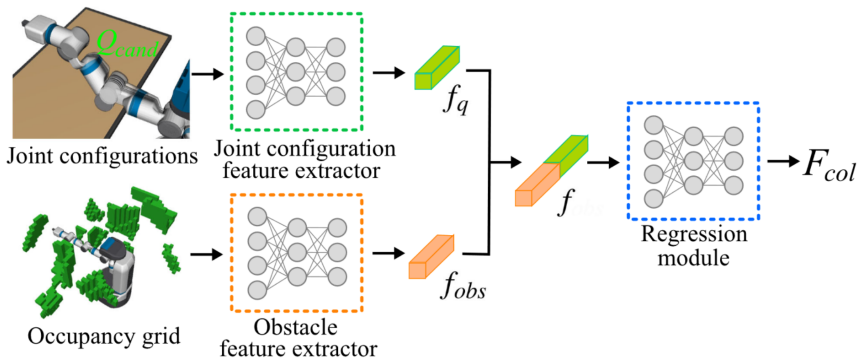


Figure 2: The collision cost prediction network of RCIK [2] is trained on simulated data, and outputs a collsion cost F_{col} which approximates the signed distance function to enable real-world real-time collision avoidance.

SDF evaluated at that 3D point. The occupancy grid can be generated in real time with one or more depth cameras. The network is trained on one million simulated examples of random environments and joint configurations. While this method does not have the collision avoidance guarantees provided in theory by other methods, it is the first method to perform collision-free inverse kinematics in real time with real sensing. This same approach was later evaluated under real-time control [49].

2.1 Future Directions in Collision Avoidance

Real-time generation of signed distance fields. Signed distance fields have proven highly effective for enabling collision avoidance. However, SDFs are very costly to generate, and require a very accurate representation of the underlying environment. Because of this, the vast majority of work on collision avoidance with robot manipulators does so only in simulation, or in manually recreated static environments. A pressing challenge is finding ways to bring these methods to the real world by constructing an SDF in real time. Recent work on neural representations has enabled real-time construction of a neural SDF from depth imagery called iSDF [50]. Similarly to RCIK [2], the SDF produced by iSDF is only an approximation, however it is generated over time from multiple sensor observations, and does not rely on simulated pre-training. Adapting a similar approach for static or manipulator-mounted depth cameras could enable real-world operation of the many collision avoidance approaches which rely on SDFs.

Moving beyond signed distance fields. Signed distance fields alone are great for collision avoidance, but offer some limitations. For example, not all collisions are equally costly. Colliding with a pillow might be admissible if it means avoiding collision with a human. Future representations, and algorithms which act on them, could store semantic information along with scene geometry, to enable such decisions to be made. Learning such semantic scene properties may be possible via extensive pre-training, or via interaction.

3. Manipulation

Arguably the most important task for a robot *manipulator* is to *manipulate* things. Manipulation can mean grasping with a simple one degree-of-freedom gripper, articulated grasping, or simple pushing and nudging of objects with any part of the robot. Robot manipulation is a vast field, with approaches specialized for dealing with many specific challenges. To keep the scope of this review reasonable, we focus on scene representations which are used for:

- Basic grasping with a 1DoF gripper
- Generalizable grasping
- Articulated grasping
- Predicting scene flow

Direct Action on Images– While this review focuses on intermediate scene representations, it is clear that, for the purposes of robot manipulation, an intermediate representation is not necessary. A seminal work in this area was Saxena et. al. [21] in 2006, which was the first work to directly predict grasping points from an image. They use a neural network to, given an RGB image of an object to be grasped, predict pixels in the image at which the object is most suitable for grasping. To train their neural network, they use supervised learning on a large fully synthetic dataset. Two years later, the same authors improved on the approach by using RGB-D imagery as input to the network [51]. Lenz et. al. [52] improved upon previous work by aiming to predict the single best grasp, rather than listing many viable ones, and predicting the orientation and extent of the grasp along with the location. Later, the "DexNet" series of papers offered iterative improvements by improving training data and tweaking the neural network outputs, as well as considering alternative grip types such as articulated hands and suction cups [53, 54, 55]. Other work aims to grasp objects given some semantic label, e.g. "coffee mug" or "plate", this is sometimes called semantic grasping. Jang et. al. [56] proposes a two-stream approach to semantic grasping, in which one stream identifies objects while another finds suitable grasps. Schwarz et. al. [57] demonstrates a semantic grasping pipeline which uses a suction cup gripper and works by simply segmenting out objects and finding their center of mass. This approach performed well at the highly competitive Amazon Picking Challenge[2]. These direct prediction approaches are highly effective for real world operation, however they are fairly limited in their potential. These approaches can only perform single-shot grasping, meaning they are unable to, for example, rotate an object, let go of it, and grab it again. They also have a limited ability to reason about novel shapes in 3D. Lastly, these approaches typically rely on some synthetic training data, which must be generated via other 3D-aware methods.

Meshes– A number of works use a 3D triangular mesh to represent objects for manipulation. The problem of finding suitable grasps given a 3D mesh is a long-standing problem with active research [58, 59, 60, 61, 62]. This paragraph focuses not on those algorithms, but on real-world systems for manipulation which represent objects as meshes. The first of such real-world systems was Berenson et. al. [63] in 2007. This work makes grasping

2. https://robohub.org/amazon-picking-challenge/

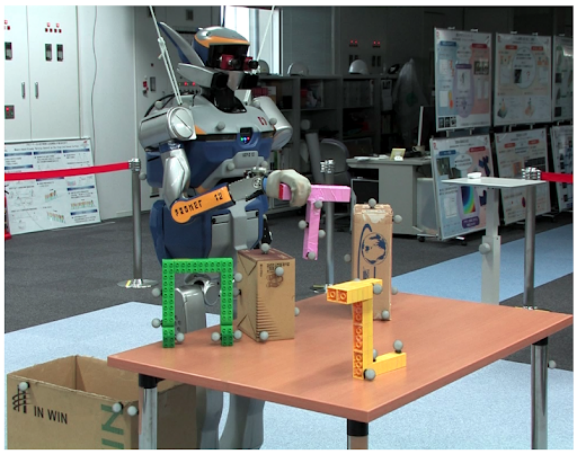
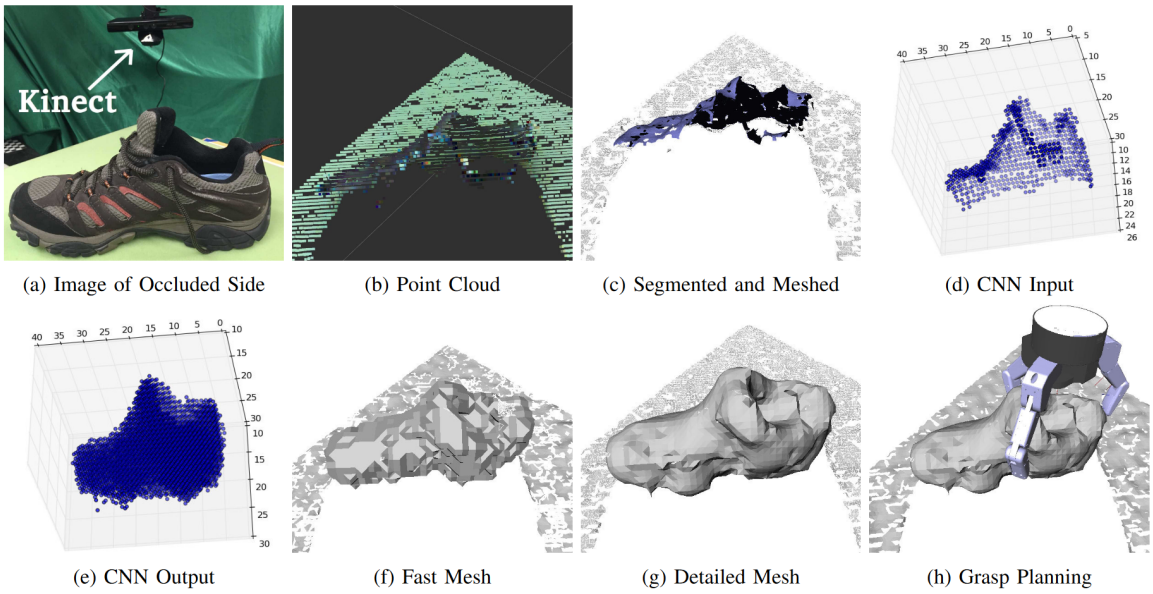


Figure 3: An early approach for real world grasping, Berenson et. al. [63], relied on a motion capture system and pre-defined meshes to perform grasping in the real world.

possible in the real world by incorporating information about not only the object to be grasped, but also nearby obstacles, such as the table or other objects, into the grasp selection algorithm. In order to sense the positions of objects in real-world tests, this approach relies on motion capture markers being placed on each object, as shown in fig. 3. Goldfeder et. al. [64] introduced a method for finding good grasps given a mesh, and tested their method by scanning real objects. This approach was somewhat effective, but the conversion from scan to mesh does not happen in real time. Collet et. al. [65] circumvents the problem of real-time mesh construction by modeling each object as a primitive, and fitting that primitive to a point cloud in real time. Later work by the same authors [66] extends this idea to arbitrary meshes by using a 6D pose recognition algorithm. Assuming that a mesh of the object is known, this approach enables prediction of the mesh's pose in real time. Papazov et. al. [67] takes a similar approach, but assumes that the object is represented with a set of points and surface normals, rather than an RGB image. Varley et. al. [68] removes the requirement of a pre-made mesh by teaching a neural network to complete a point cloud. From the completed point cloud, a mesh can be built and that mesh passed off to a mesh-based grasp planner in real time. The steps in Varley et. al.'s pipeline are shown in fig. 4. In each of these approaches, we see that the limiting factor is not the grasp planning algorithms themselves, but the generation of meshes in real time.

Point Clouds– Point clouds have been used as the basis for grasp planning in a similar manner to meshes. In constrast to meshes, point clouds are closer to the native output of commonly used depth cameras, making them more practical for real world construction. Approaches exist for grasp planning directly on point clouds [69, 70, 71, 72], although less numerous than those for meshes. Florence et. al. [73] (covered in more depth in the following paragraph) introduces a method for finding corresponding grasps between similar objects, and uses point clouds to perform the grasping in real-world examples. Simeonov et. al. [74] considers the problem of manipulation planning, and is able to predict the movement of scene objects directly from their point clouds. This prediction is used to plan for manipulations which move the scene towards a goal state.



(a) Image of Occluded Side (b) Point Cloud (c) Segmented and Meshed (d) CNN Input

(e) CNN Output (f) Fast Mesh (g) Detailed Mesh (h) Grasp Planning

Figure 4: Varley et. al. [68] uses a neural network to enable shape completion on partial point cloud observations. The completed shapes can then be transformed to a mesh and used in any mesh-based grasp planner. Image from Varley et. al.

Voxel Grids– Zhang et. al. [75] constructs a gripper-sized voxel grid from a point cloud and a given gripper position, with each voxel encoding whether the space is occupied by a point in the point cloud. This voxel grid can then be compared (via nearest neighbors) to previously simulated voxel grids to determine whether it corresponds to a good gripper position.

Learned Representations– Dense object nets [73] are an approach for generalizable grasping. For a given input image, a neural network is trained to produce a dense pixel-level feature map which produces similar output feature vectors for semantically similar points in multiple images of similar objects. For example, for any two images of a mug, the goal is that pixels corresponding to the same point on the mug handle will have the same feature vector in the output representation. This representation is trained to be consistent across object instance, pose, and deformation, enabling generalizable grasping. They demonstrate that this approach is effective at generalizable grasping in the real world, using merged point cloud data from multiple depth cameras. Simeonov et. al. [19] expands upon this approach by taking a 3D point as input to the neural network, rather than a 2D pixel position. The object properties are encoded within the weights of a multi-layer perceptron, similarly to NeRF [45]. They demonstrate that this approach can be used to perform pick-and-place tasks of novel objects of a given class given fewer than 10 demonstrations. Additionally, the architecture of their MLP ensures that the descriptor fields are SE(3)-equivariant, making them robust to arbitrary object poses. Van Der Merwe et. al. [76] uses a learned representation for articulated grasping. A neural network is trained to, given a point cloud and a 3D query point, approximate the signed distance function at that point. The latent space of this network is concatenated with information about desired grasp qualities and robot

configuration, and fed into another network which predicts the success rate of the proposed grasp. Xu et. al. [20] builds a visual predictive model for robotics. Their goal is to, given some representation of the scene along with a robot action, predict the scene after the action has occurred. Rather than using a manually constructed scene representation, they allow the scene representation to be learned in an end-to-end manner, as a 128x128x48 feature grid with an 8-dimensional vector at each point in the grid. Using this representation, they are able to predict 3D scene flow and plan manipulation tasks.

3.1 Future Directions in Manipulation

Performing 3D grasp planning based on real-world sensor data. There is a disconnect between the way scenes are represented for grasp planning (primarily meshes) and the way that the most effective robotics systems, such as those used in the Amazon Picking Challenge, are operated (primarily direct prediction). An important direction for future work is bridging this gap. Direct prediction methods are severely limited in their spatial reasoning, while mesh-based methods are severely limited in their real-world operation. Point cloud manipulation and learned representations may bridge the gap, but do not currently offer the high grasping accuracy of mesh based approaches.

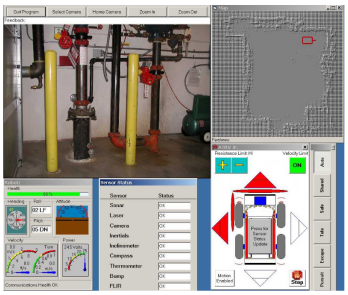
Representing complex object properties. Any object may have many properties which are relevant to manipulation: its center of mass, deformation properties, or constraints on its motion (e.g. hold a mug full of coffee upright, pull a drawer straight out). Learning-based approaches, such as Dense PhysNet [77] have shown promise towards being able to learn these properties autonomously. A needed direction for future research is determining ways to learn these properties, generalize them to novel objects, and store these properties alongside their geometric representations.

4. Teleoperation

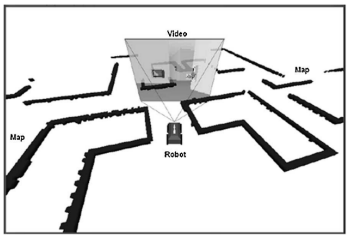
In a teleoperation scenario, a human controls a robot remotely, and relies on an intermediate interface, such as a monitor or VR headset, to understand the robot's surroundings. Much recent work on teleoperation places the human operator in virtual reality (VR); studies have shown increased task performance in virtual reality, due to the ease of controlling the user's view and all six degrees-of-freedom of the robot [78]. However, the scene representations used in these VR approaches can generally applied to any sort of teleoperation scenario.

4.1 Human Control of Mobile Robots

Mobile robots are much more likely than robot manipulators to venture into unknown environments. Because of this, the foundational work in representing scenes to remote operators has taken place in mobile robotics. For context, we offer a brief overview of such work here. Nielsen et. al. [79] was an early attempt at incorporating data beyond RGB camera feeds into robot teleoperation. They create a dashboard which shows an RGB camera feed along with LiDaR scans and a rough map of the environment. While all of the information is presented to the user, it is not combined to create an intuitive display. Kelly et. al. [80] builds a 3D map of the environment by using LiDaR scans to create a 3D map, and coloring that map with data from RGB images. A CAD model of the mobile robot is



(a) Dashboard-based interface from [82] (2004)



(b) Augmented interface from [79] (2007)

(c) 3D interface from [80] (2011)

Figure 5: A history of the typical information displays used in mobile robot teleoperation

placed in the environment. This allows rendering arbitrary viewpoints, such as an overview of the entire scene, an overhead view, or a third person view, as would be seen in a racing video game. Stotko et. al. [81] builds a 3D mesh-based scene representation for a mobile manipulator, and displays the mesh to the user interactively in virtual reality. They find that users have fewer collisions, and report a greater level of immersion and awareness than with a 2D interface. Livantino et. al. [23] augments robot-attached camera views with 3D data to overlay data such as desired path, destination, and label traversable terrain. For mobile robot teleoperation, the trend has been towards a free floating, user controllable view and natural display of information. These same goals apply to robot manipulator teleoperation.

4.2 Human Control of Robot Manipulators

No intermediate representation– A simple baseline for teleoperation is the use of one or more static cameras, which the user can either see all at once, or switch between manually. While simple to implement, a static camera approach is limiting: they present the user with limited geometric information, and don't leverage computation to enhance human perception. Static cameras are also prone to being blocked by the robot manipulator itself. There are a number of approaches which improve on the static camera baseline without building an intermediate representation. Murata et. al. [83] considers mobile manipulators (manipulators mounted on mobile robots), and renders a CAD model of the robot on top of a background made of stitched, observed images. The CAD model is kept updated to represent the robot's current state, and the user is able to position the camera to generate an arbitrary view. A different approach is taken by Rakita et. al. [84], in which a robot arm is used as a camera operator for another robot arm. The camera operator's pose is automatically optimized in real time to present a clear view of the manipulating robot's end effector.

Point Clouds– Point clouds are a common choice for representing scenes for real time operation, because they are the native output of depth cameras, and can be displayed in real time with little processing. Brizzi et. al. [85] considers augmented reality for VR teleoperation. Operators see an RGB image of the scene, along with features, such as distance to the target, direction to the target, or gripper state. Point clouds from a depth

Figure 6: Kohn et. al. [86] uses meshes to represent known objects (table and robot) and point clouds to represent unknown objects (puzzle box) for teleoperation. Image from Kohn et. al.

camera are used to calculate these features. Kohn et. al. [86] displays a combination of point clouds from RGBD cameras and meshes to a user in VR. Meshes are used for known objects (such as the floor, table, and robot) and point clouds are used for the unknown. The unknown objects are filtered in real time according to the meshes, as shown in fig. 6. A similar approach is taken by Su et. al. [87]. Wei et. al. [22] similarly displays point clouds to users, but unlike previous approaches, they do not use meshes to represent known objects, aside from the robot gripper. They perform a user study, comparing the point cloud to multi-view RGB and to a point cloud projected onto an RGB image. In their experiments, the hybrid point cloud and RGB view performs best.

Meshes– While point clouds are native and fast to display, meshes may be preferred due to their potentially higher visual fidelity. Wonsick et. al. [88] takes an approach similar to the mesh-based approaches in section 3; a point cloud of the scene is semantically segmented to identify its class amongst known objects. A known 3D model is then fit to each point cloud segment, and that 3D model is displayed to the user in virtual reality. Models are able to be constructed in much higher fidelity, and the scene can be entirely rendered, enabling control of lighting, texture, etc. They run a user study and say that users find their approach more usable and less cognitive load than point-cloud streaming alone. However their approach relies on having known 3D models of objects in the environment, and has no fallback if such a model does not exist, as such it is not suitable to unknown environments. Piyavichayanon et. al. [89] generates a "depth mesh" by combining the view of multiple depth cameras. This mesh is used to display augmented reality features, such as distance to a collision state, on a handheld smartphone. Merwe et. al. [90] performs a user study to investigate how the type of information presented to the operator changes their performance. They compare a full information 3D model to a "representative" mesh based model, which only displays crucial information. They find that task completion time is lower with the full model, but cognitive load is higher.

Occupancy Grids– Omarali et. al. [91] considers multiple modes of visualization for VR teleoperation. Alongside the depth camera-based baselines, they present a hybrid view, which displays the current output from the depth camera, alongside a translucent occupancy map in the previously observed areas. This occupancy map gives the user some context for the greater environment, while indicating that the context is not as reliable as the currently

observed region. They find that users prefer the hybrid depth camera + occupancy map approach.

4.3 Future Directions in Teleoperation

Improving real-time visualization of the robot's environment. Existing approaches tend to rely on depth cameras, which can render the world in real time, but are low in detail and high in noise. Approaches which attempt to process this data into something more appealing, like a mesh, require heavy computation and require meshes of potential objects to be known a priori. Future work should consider ways to improve the fidelity and detail of reconstruction in real time and in unknown environments. To some extent, advances in imaging, such as high resolution depth sensors may improve this problem. Another potential solution is dynamically directing sensing towards areas that need high detail, such as around the end effector. The problem of representing the environment from a novel perspective is known as novel view synthesis and is well studied in the computer vision literature (see the recent review [92]). Incorporating techniques from novel view synthesis could allow a more complete environment representation to be displayed to the operator.

Building representations suited to the operator. Currently, the vast majority of approaches focus on building representations which represent the environment faithfully. However, it has long been known that realistic displays are not always ideal, as they can make it difficult to parse what is relevant [93]. Some work covered in this review has considered building a sparse representation of only relevant features [90, 88]. However, these approaches require such a representation to be specified by hand prior to operation. Future work should consider ways of filtering the scene representation to represent only what is important to the user in an unknown scene.

5. Future Directions

Quantifying Uncertainty. Present systems for robot manipulation build a *most likely* map of the environment. In poor sensing conditions, this map may be highly unreliable, leading the downstream robotics application to make incorrect but fully confident choices based on the unreliable map. Future work should work to incorporate uncertainty into the scene representation itself, in order to enable robot policies which act intelligently in the face of uncertainty. When in a highly uncertain environment such a robot could, for example, ask a human for help, or gather additional information about its environment before taking action. The issue of uncertainty estimation is well studied in mobile robotics, especially in the context of SLAM [94]. Recent approaches in novel view synthesis have also enabled dense estimates for geometric and photometric uncertainty [95, 96], as shown in fig. 7. Such approaches could be applied to robot manipulators, especially for problems like collision avoidance.

Joint prediction of scene semantics and geometry. Present approaches which infer scene semantics do so by taking some geometric representation of the scene as input, and producing a semantic map based on that geometry. With this approach, scene geometry informs scene semantics, but semantics has no influence on geometry. In reality, semantic information about an object can provide queues about its likely geometry. If an object is identified as a coffee mug, it probably has a handle on it. If an object is identified as a

Figure 7: Shen et. al. [95] builds a scene representation which includes dense uncertainty estimates. A similar approach could prove useful to build scene representations for robot manipulators. Image from Shen et. al.

baseball, it is probably spherical. Floors and tabletops tend to be flat and level with one another. Future approaches could infer scene geometry and semantics jointly, by encoding sensor data in a learned latent representation, which is used to inform both geometric and semantic inference. Joint inference approaches have been used for semantic SLAM [97, 98, 99]. Implementing such an approach for robot manipulators would enable higher fidelity predictive mapping and improved semantic understanding.

Sensing-first development and real-world benchmarking. Many existing approaches, particularly for grasping and collision avoidance, are benchmarked entirely in simulation. Real-world demonstrations are addressed after the development has been optimized for simulation, and often only occur under highly controlled conditions. Future work should consider developing methods with sensing in mind from the beginning, and aiming for high performance on real-world tasks with sensing, rather than in simulation. This means building systems which are robust to sensor noise, and able to act on raw sensor data in real time. One encouraging sign towards such a goal is the just announced RT-1 project from Google [100], which is an embodied agent that is benchmarked entirely on real-world tasks.

Representations for Alternative Sensing Modalities. All works addressed in this survey utilize data from typical vision based sensors: RGB cameras, depth cameras, and/or LiDaR. There are many other sensor types used in robotics, such as force torque sensors and tactile sensors, which provide information about the environment. However, existing approaches react immediately to information from these sensors, and do not build an intermediate representation which persists over time. Can we build scene representations which model the factors that these sensors measure? For example, the measurement from a tactile sensor may be influenced by how hard the object is, how heavy it is, or even how conductive it is, alongside the object's geometry. Present scene representations do not encode all of this information, making it very difficult to relate new observations from a tactile sensor to the known scene. Creating scene representations which do encode information from alternative sensors could enable applications such as SLAM or collision avoidance under low vision conditions.

6. Conclusion

Building new scene representations for robot manipulation is an important step towards creating fully autonomous embodied agents which can interact intelligently with the world.

Continued advances in sensing and robotics will necessitate new representations which encode data from new sensors and support new robot form factors and applications. There are two challenges which are present in all works covered in this review: building representations which can be constructed in real time, and which are robust to sensor noise. Given the recent pace of advancement in robotics and computing, we are confident that these challenges will be sufficiently overcome.

Application	**Common Representations**	**Less Common Representations**
Collision Avoidance	Signed distance fields	Convex hulls Potential fields Geometric primitives Learned representations
Manipulation and Grasping	Direct action Meshes Point clouds	Voxel grids Signed Distance Fields Learned representations
Teleoperation	Point clouds	Meshes Occupancy grids

Table 2: Scene representations covered in this review

References

[1] Z. Moore, C. Sifferman, S. Tullis, M. Ma, R. Proffitt, and M. Skubic, "Depth sensor-based in-home daily activity recognition and assessment system for stroke rehabilitation," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 1051–1056.

[2] M. Kang, Y. Cho, and S.-E. Yoon, "Rcik: Real-time collision-free inverse kinematics using a collision-cost prediction network," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 610–617, 2022.

[3] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 489–494.

[4] C. Landsiedel, V. Rieser, M. Walter, and D. Wollherr, "A review of spatial reasoning and interaction for real-world robotics," *Advanced Robotics*, vol. 31, no. 5, pp. 222–242, 2017.

[5] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 1, jul 2022.

[6] Y. Cong, R. Chen, B. Ma, H. Liu, D. Hou, and C. Yang, "A comprehensive study of 3-d vision-based robot manipulation," *IEEE Transactions on Cybernetics*, pp. 1–17, 2021.

[7] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aat8414

[8] J. Cui and J. Trinkle, "Toward next-generation learned robot manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd9461, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abd9461

[9] F. Guerin and P. Ferreira, "Robot manipulation in open environments: New perspectives," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 3, pp. 669–675, 2020.

[10] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, 2018.

[11] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 215–242, 2021. [Online]. Available: https://doi.org/10.1146/annurev-control-072720-082553

[12] J. Fan, P. Zheng, and S. Li, "Vision-based holistic scene understanding towards proactive human–robot collaboration," *Robotics and Computer-Integrated*

Manufacturing, vol. 75, p. 102304, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584521001848

[13] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[14] M. Wonsick and T. Padir, "A systematic review of virtual reality interfaces for controlling and interacting with robots," *Applied Sciences*, vol. 10, no. 24, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/24/9051

[15] Y. K. Hwang and N. Ahuja, "Gross motion planning—a survey," *ACM Comput. Surv.*, vol. 24, no. 3, p. 219–291, sep 1992. [Online]. Available: https://doi.org/10.1145/136035.136037

[16] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016.

[17] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13310

[18] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9995–10 001.

[19] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se(3)-equivariant object representations for manipulation," *CoRR*, vol. abs/2112.05124, 2021. [Online]. Available: https://arxiv.org/abs/2112.05124

[20] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3d dynamic scene representations for robot manipulation," in *Conference on Robot Learning (CoRL)*, 2020.

[21] A. Saxena, J. Driemeyer, J. Kearns, and A. Ng, "Robotic grasping of novel objects," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19. MIT Press, 2006. [Online]. Available: https://proceedings.neurips.cc/paper/2006/file/22722a343513ed45f14905eb07621686-Paper.pdf

[22] D. Wei, B. Huang, and Q. Li, "Multi-view merging for robot teleoperation with virtual reality," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8537–8544, 2021.

[23] S. Livatino, D. C. Guastella, G. Muscato, V. Rinaldi, L. Cantelli, C. D. Melita, A. Caniglia, R. Mazza, and G. Padula, "Intuitive robot teleoperation through multi-sensor informed mixed reality visual aids," *IEEE Access*, vol. 9, pp. 25 795–25 808, 2021.

[24] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[25] G. Buizza Avanzini, N. M. Ceriani, A. M. Zanchettin, P. Rocco, and L. Bascetta, "Safety control of industrial robots based on a distributed distance sensor," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2127–2140, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6740805

[26] S. Tsuji and T. Kohama, "Proximity skin sensor using time-of-flight sensor for human collaborative robot," *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5859–5864, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8669855

[27] S. M. LaValle. Cambridge University Press, 2006.

[28] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[29] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[30] F. Miyazaki, S. Arimoto, M. Takegaki, and Y. Maeda, "Sensory feedback based on the artificial potential for robot manipulators," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 2381–2386, 1984.

[31] V. Pavlov and A. Voronin, "The method of potential functions for coding constraints of the external space in an intelligent mobile robot," *Soviet Automatic Control*, vol. 17, no. 6, pp. 45–51, 1984.

[32] S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[33] S. Frisken, R. Perry, A. Rockwood, and T. Jones, "Adaptively sampled distance fields: A general representation of shape for computer graphics," *ACM SIGGRAPH*, 07 2000.

[34] S. Gibson, "Using distance maps for accurate surface representation in sampled volumes," in *IEEE Symposium on Volume Visualization (Cat. No.989EX300)*, 1998, pp. 23–30.

[35] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4569–4574.

[36] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: science and systems*, vol. 9, no. 1. Citeseer, 2013, pp. 1–10.

[37] C. Park, J. Pan, and D. Manocha, "Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 22, no. 1, pp. 207–215, May 2012. [Online]. Available: https://ojs.aaai.org/index.php/ICAPS/article/view/13513

[38] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.

[39] L. Han, F. Gao, B. Zhou, and S. Shen, "FIESTA: fast incremental euclidean distance fields for online motion planning of aerial robots," *CoRR*, vol. abs/1903.02144, 2019. [Online]. Available: http://arxiv.org/abs/1903.02144

[40] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *IJCAI*, 2015.

[41] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter, "Collision-free MPC for legged robots in static and dynamic scenes," *CoRR*, vol. abs/2103.13987, 2021. [Online]. Available: https://arxiv.org/abs/2103.13987

[42] S. Zimmermann, M. Busenhart, S. Huber, R. Poranne, and S. Coros, "Differentiable collision avoidance using collision primitives," 2022. [Online]. Available: https://arxiv.org/abs/2204.09352

[43] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, p. 469–483, dec 1996. [Online]. Available: https://doi.org/10.1145/235815.235821

[44] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014. [Online]. Available: https://doi.org/10.1177/0278364914528132

[45] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[46] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *arXiv preprint arXiv:1906.07751*, 2019.

[47] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[48] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022.

[49] M. Kang, M. Yoon, and S.-E. Yoon, "User command correction for safe remote manipulation in dynamic environments," *Human-Robot Interaction (HRI)*, 2022.

[50] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "isdf: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[51] A. Saxena, L. L. S. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *AAAI Conference on Artificial Intelligence*, 2008.

[52] 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.

[53] 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*, vol. abs/1703.09312, 2017.

[54] J. Mahler, M. Matl, X. Liu, A. Li, D. V. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning," *CoRR*, vol. abs/1709.06670, 2017. [Online]. Available: http://arxiv.org/abs/1709.06670

[55] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aau4984

[56] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," *CoRR*, vol. abs/1707.01932, 2017. [Online]. Available: http://arxiv.org/abs/1707.01932

[57] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3347–3354.

[58] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, *Physics-based grasp planning through clutter*. MIT Press, 2012.

[59] S. Duenser, J. M. Bern, R. Poranne, and S. Coros, "Interactive robotic manipulation of elastic objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3476–3481.

[60] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 1710–1716.

[61] F. T. Pokorny, K. Hang, and D. Kragic, "Grasp moduli spaces," in *Robotics: Science and Systems*, 2013.

[62] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 557–562.

[63] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007, pp. 42–48.

[64] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1710–1716.

[65] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 48–55.

[66] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, pp. 1284 – 1306, 2011.

[67] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 538–553, 2012. [Online]. Available: https://doi.org/10.1177/0278364911436019

[68] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," 2016. [Online]. Available: https://arxiv.org/abs/1609.08546

[69] 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. [Online]. Available: https://doi.org/10.1177/0278364917735594

[70] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3619–3625.

[71] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, and X. Shen, "Robotics dexterous grasping: The methods based on point cloud and deep learning," *Frontiers in Neurorobotics*, vol. 15, p. 73, 2021.

[72] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," in *Robotics research*. Springer, 2018, pp. 307–324.

[73] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," 2018. [Online]. Available: https://arxiv.org/abs/1806.08756

[74] A. Simeonov, Y. Du, B. Kim, F. R. Hogan, J. B. Tenenbaum, P. Agrawal, and A. Rodriguez, "A long horizon planning framework for manipulating rigid pointcloud objects," *CoRR*, vol. abs/2011.08177, 2020. [Online]. Available: https://arxiv.org/abs/2011.08177

[75] L. Zhang, "Grasp evaluation with graspable feature matching," 2010.

[76] M. V. der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning Continuous 3D Reconstructions for Geometrically Aware Grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. [Online]. Available: https://sites.google.com/view/reconstruction-grasp/home

[77] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "Densephysnet: Learning dense physical object representations via multi-step dynamic interactions," 2019. [Online]. Available: https://arxiv.org/abs/1906.03853

[78] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, "Comparing robot grasping teleoperation across desktop and virtual reality with ros reality," in *Robotics Research*. Springer, 2020, pp. 335–350.

[79] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, 2007.

[80] A. Kelly, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, and E. Capstick, "Real-time photorealistic virtualized reality interface for remote mobile robot control," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 384–404, 2011.

[81] P. Stotko, S. Krumpen, M. Schwarz, C. Lenz, S. Behnke, R. Klein, and M. Weinmann, "A vr system for immersive teleoperation and live exploration with a mobile robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3630–3637.

[82] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition," *Human–Computer Interaction*, vol. 19, no. 1-2, pp. 117–149, 2004.

[83] R. Murata, S. Songtong, H. Mizumoto, K. Kon, and F. Matsuno, "Teleoperation system using past image records for mobile manipulator," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4340–4345.

[84] D. Rakita, B. Mutlu, and M. Gleicher, "An autonomous dynamic camera method for effective remote teleoperation," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 325–333.

[85] F. Brizzi, L. Peppoloni, A. Graziano, E. Di Stefano, C. A. Avizzano, and E. Ruffaldi, "Effects of augmented reality on the performance of teleoperated industrial assembly tasks in a robotic embodiment," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 2, pp. 197–206, 2017.

[86] S. Kohn, A. Blank, D. Puljiz, L. Zenkel, O. Bieber, B. Hein, and J. Franke, "Towards a real-time environment reconstruction for vr-based teleoperation through model segmentation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[87] Y. Su, Y. Xu, S. Cheng, C. Ko, and K.-Y. Young, "Development of an effective 3d vr-based manipulation system for industrial robot manipulators," in *2019 12th Asian Control Conference (ASCC)*. IEEE, 2019, pp. 1–6.

[88] M. Wonsick, T. Keleştemur, S. Alt, and T. Padır, "Telemanipulation via virtual reality interfaces with enhanced environment models," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2999–3004.

[89] C. Piyavichayanon, M. Koga, E. Hayashi, and S. Chumkamon, "Collision-aware ar telemanipulation using depth mesh," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2022, pp. 386–392.

[90] D. B. Merwe, L. V. Maanen, F. B. T. Haar, R. J. Van Dijk, N. Hoeba, and N. V. d. Stap, "Human-robot interaction during virtual reality mediated teleoperation: How environment information affects spatial task performance and operator situation awareness," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 163–177.

[91] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle, and I. Farkhatdinov, "Virtual reality based telerobotics framework with depth cameras," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 1217–1222.

[92] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "Nerf: Neural radiance field in 3d vision, a comprehensive review," *arXiv preprint arXiv:2210.00379*, 2022.

[93] H. S. Smallman and M. F. S. John, "Naive realism: Misplaced faith in realistic displays," *Ergonomics in Design: The Quarterly of Human Factors Applications*, vol. 13, pp. 13 – 6, 2005.

[94] M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active slam," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 829–834, 2018.

[95] J. Shen, A. Ruiz, A. Agudo, and F. Moreno-Noguer, "Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 972–981.

[96] N. Sünderhauf, J. Abou-Chakra, and D. Miller, "Density-aware nerf ensembles: Quantifying predictive uncertainty in neural radiance fields," 2022. [Online]. Available: https://arxiv.org/abs/2209.08718

[97] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[98] N. Sünderhauf, F. Dayoub, S. McMahon, M. Eich, B. Upcroft, and M. Milford, "Slam-quo vadis? in support of object oriented and semantic slam," in *Proceedings of the RSS 2015 Workshop-Problem of mobile sensors: Setting future goals and indicators of progress for SLAM*. Australian Centre for Robotic Vision, 2015, pp. 1–7.

[99] K. Doherty, D. Fourie, and J. Leonard, "Multimodal semantic slam with probabilistic data association," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 2419–2425.

[100] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," 2022. [Online]. Available: https://arxiv.org/abs/2212.06817