Multimodal Technologies and Interaction

MDPI

Perspective

Keep the Human in the Loop: Arguments for Human Assistance in the Synthesis of Simulation Data for Robot Training

Carina Liebers [1,*], Pranav Megarajan [2], Jonas Auda [1], Tim C. Stratmann [2], Max Pfingsthorn [2], Uwe Gruenefeld [1] and Stefan Schneegass [1]

[1] HCI Group, Faculty of Computer Science, University of Duisburg-Essen, Schuetzenbahn 70, 45127 Essen, Germany; jonas.auda@uni-due.de (J.A.); uwe.gruenefeld@uni-due.de (U.G.); stefan.schneegass@uni-due.de (S.S.)
[2] OFFIS—Institute for IT, 26121 Oldenburg, Germany; tim.stratmann@offis.de (T.C.S.)
* Correspondence: carina.liebers@uni-due.de

Abstract: Robot training often takes place in simulated environments, particularly with reinforcement learning. Therefore, multiple training environments are generated using domain randomization to ensure transferability to real-world applications and compensate for unknown real-world states. We propose improving domain randomization by involving human application experts in various stages of the training process. Experts can provide valuable judgments on simulation realism, identify missing properties, and verify robot execution. Our human-in-the-loop workflow describes how they can enhance the process in five stages: validating and improving real-world scans, correcting virtual representations, specifying application-specific object properties, verifying and influencing simulation environment generation, and verifying robot training. We outline examples and highlight research opportunities. Furthermore, we present a case study in which we implemented different prototypes, demonstrating the potential of human experts in the given stages. Our early insights indicate that human input can benefit robot training at different stages.

Keywords: robot training; human-in-the-loop; simulation; machine learning

check for updates

Citation: Liebers, C.; Megarajan, P.; Auda, J.; Stratmann, T.C.; Pfingsthorn, M.; Gruenefeld, U.; Schneegass, S. Keep the Human in the Loop: Arguments for Human Assistance in the Synthesis of Simulation Data for Robot Training. *Multimodal Technol. Interact.* **2024**, *8*, 18. https://doi.org/10.3390/mti8030018

Academic Editor: Michael Gienger

Received: 22 November 2023
Revised: 15 February 2024
Accepted: 20 February 2024
Published: 1 March 2024

1. Introduction

Real-world simulations are used to train robots so that they can transfer their behavior to the physical world, known as simulation-to-real (sim2real) [1]. An example is training a vacuum cleaner robot in various virtual simulation environments before placing it in a real-world setting, allowing it to learn to clean efficiently and avoid obstacles in a safe and controlled virtual environment. However, if the training does not include all possible real-world states, it can lead to incorrect behavior [1]. Thus, most sim2real techniques utilize domain randomization (DR) to generate myriads of simulated environments with randomized properties such as object location, rotation, and color. They offer enough variability that even unrepresented real-world states can appear to the robot as other learned variations, making it robust in new situations [1,2]. This allows a robot to execute tasks despite encountering objects with different colors or placements [2]. In short, DR enables robot applications to achieve adequate accuracy when deployed in the real world [2].

In contrast, narrowing the application scenario was shown to improve training performance by generating application-specific simulation environments [3]. Eliminating virtual states that the robot will never see in the real world means the robot does not have to learn as many scenarios, expediting the training process. To select relevant states, context-focused approaches cover the complexity of the real world with meaningful and diverse parameter distributions [4], select object positions with task-specific probability functions [3], or contextual information about the interference of environmental properties [5]. However, despite the promise of these methods, they primarily emphasize algorithmic and unsupervised learning strategies, often overlooking the substantial contributions that human expertise can bring to the table. This oversight represents a significant missed opportunity. Human knowledge is, after all, not only invaluable during the development phase but also allows end users, who are experts in their respective application domains, to offer meaningful insights. Therefore, integrating this human-centric perspective could greatly enhance the effectiveness and relevance of these approaches.

In recent years, human-centered approaches have revealed ways to utilize human perception and domain knowledge to enhance related approaches and artifacts. For instance, humans have assisted with object labeling [6], point cloud part selection, and labeling [7,8] and interactive segmentation using touch and voice commands [9] in Virtual Reality (VR). Thus, regarding robot training, we envision that end users can easily assist in specifying object existence, spatial movability, or material properties to enhance current approaches to generating realistic simulation environments. As they are experts in the application context, we refer to them as *application experts*.

In this work, we propose a conceptual workflow that distinguishes five sequential stages of human intervention in robot training: (1) validating and improving real-world scans, (2) correcting virtual representations, (3) specifying application-specific object properties, (4) verifying and influencing simulation environment generation, and (5) verifying robot training. At each stage, we introduce opportunities for application experts to collaborate in enhancing the simulation environment generation for robot training. Building on these stages, we conducted a case study in which we developed prototypes for the individual stages. Thereby, we aim to demonstrate the feasibility of our concept as a whole and our individual proposed ideas to improve training. Thus, our work contributes the following:

(1) A conceptual workflow of five stages that identifies opportunities for involving application experts in robot training, particularly for the generation of simulation environments (Section 3).

(2) A case study in which we implemented different prototypes for the individual stages to gain insights into their feasibility and to illustrate our set of proposed ideas for improving robot training by keeping the human in the loop (Section 4).

2. Related Work

Our concept aims to reduce the number of simulation environments required to realistically represent real-world scenes by involving application experts. Since we focus on human-in-the-loop simulation environment generation, all research addressing human engagement in the individual stages and generation of simulation environments for human-robot interaction is relevant.

2.1. Robot Training

To create virtual representations of real-world environments, sensors capture the physical surroundings. Using their data, virtual representations are generated, mostly as point clouds [10]. To map the scan data to semantically meaningful objects in the real-world scene, pre-processing is often necessary [11,12]. Next, virtual simulations of the environment can be designed. This requires specification of the appropriate parameters of the object's distribution, which can be tedious and requires expert knowledge [13]. Thus, approaches such as Automatic Domain Randomization (ADR) automate the generation of complex scenarios for training [14–17]. However, multiple randomized parameters such as textures, shapes, and colors add complexity, causing the robot to learn multiple policies. From these, developers must select the best [18]. Approaches based on Imitation Learning (IL) also use simulation environments for learning purposes. Data are collected via demonstrations of tasks in the simulation environment generated [19–21]. In some cases, domain randomization is utilized to synthesize massive numbers of demonstrations from as few human demonstrations as possible [22].

2.2. Human Engagement in Simulation Environment Generation

Although mapping scan data to semantically meaningful objects is often performed using 2D interfaces, VR was found to be promising due to its 3D environment, which makes navigation easier [8]. Using VR, prior works proposed various methods for object labeling using boundary boxes [6], point cloud part selection and labeling throughout recorded video streams [7], or a tablet to enhance the selection accuracy for fine-grain adjustments [23]. Users can further interactively segment point clouds by touching objects or surfaces and verbally identifying them to trigger algorithms to reconstruct 3D models of the selected objects [9]. Some libraries offer visualization of the physical properties overlaying a 3D mesh while maintaining point cloud shades and real-time interference of random walk properties of recorded trajectories, aiding in the analysis of point cloud data [24].

Although we are not aware of any work that enables application experts to influence simulation environment generation, approaches exist that enable non-experts to interact with and modify environmental scans. Such scans of environments can be used to create multiple versions of the environment [25] or make it interactive [26]. Wang et al. proposed a workflow for remote content creation in which the environment is scanned and can be viewed on site [25]. Furthermore, VRFromX allows users to replace real-world objects with virtual models and add functionalities to create interactive VR scenes [26].

2.3. Mixed Reality Approaches in Human-Robot Interaction

Mixed Reality (MR) applications are increasingly being applied to human-robotics interaction (HRI), mostly for teleoperation, simulation, or explaining robot actions [27]. These applications enable humans without a robotics background to use their knowledge of the application environment to expand the robot's operating range [28], provide a path [29], and validate automated systems using live visualization of simulation environments with human actions [30]. While these approaches address the interaction between robots and humans, our concept explores how application experts can influence robot training. Specifically, we examine how application experts can specify information and determine its usage to generate simulation environments that represent possible real-world scenes.

3. Human-Assisted Simulation-Based Robot Training

To enhance domain randomization methods for sim2real approaches, we developed the human-assisted workflow in two stages. First, we analyzed the existing research using the sim2real approach [3,18,31,32] and derived a generic simulation-based robot training workflow without application expert supervision (see Figure 1, gray cycle). We were interested in the essential stages of a universal workflow applicable to simulation-based robot training using real-world scans. Overall, we identified four relevant stages, described in Section 3.1. Second, we identified opportunities where application experts could enhance the stages to reduce the number of simulations to only those that fit the final application context in an iterative process and extend the specified workflow (see Figure 1, blue cycle). We discuss these opportunities in Section 3.2 and outline promising research opportunities for the presented stages.

3.1. Classical Workflow for Generating Simulation Environments without Application Expert Involvement

Based on prior work [3,18,31,32], we identified four main stages to create simulation environments from sensor scans using the sim2real approach. These encompass scanning the real-world scene (Figure 1a), computing virtual representation(s) (Figure 1b), generating simulation environments (Figure 1d), and robot training (Figure 1e).

3.1.1. Scanning the Real-World Scene (Figure 1a)

All publications covering real-world scans first involve recording the environment [18,31]. Thus, our workflow starts with the robot sensing the environment. Sensors are often directly mounted on robots, allowing them to perceive and capture the environment. The data gathered in this stage represent the scanned environment, commonly using point clouds [10].

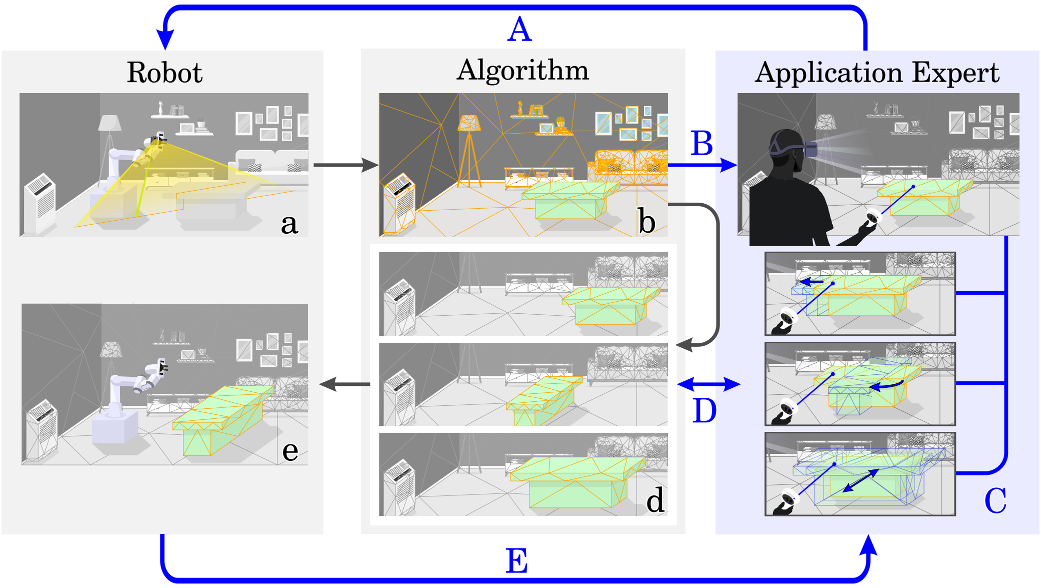

Figure 1. Without application experts (gray cycle), current workflows involve scanning the real-world scene (**a**), creating virtual representations (**b**), generating simulation environments (**d**), and training robots (**e**). We propose application experts to assist (blue cycle) in this process by validating or enhancing scans (A), refining virtual representations (B), specifying object properties (C), influencing simulation environment generation (D), and validating robot training (E).

3.1.2. Computing Virtual Representation(s) (Figure 1b)

The sensor data is used to create a virtual representation of the robot's environment. To reconstruct all virtual objects [10,33], it needs to be processed, often encompassing various methods such as segmentation, filtering, classification, and mesh generating algorithms [31].

3.1.3. Generating Simulation Environments (Figure 1d)

Given all objects in a scene, the object instances can be modified to generate simulation environments. This stage involves creating multiple variants of the simulation environment using DR [3,32]. DR can randomize object positions, rotations, sizes, and textures. It generates multiple simulation environments characterized by the variance of these properties to enable a robot agent to generalize its training and, thereby, respond appropriately when confronted with unknown scenarios [2].

3.1.4. Robot Training (Figure 1e)

Finally, the robot agent learns expected behavior using either Reinforcement Learning (RL) methods [18,31,34,35] by training in simulated environments or Imitation Learning (IL) by copying demonstrations collected in simulation environments [19,20,22]. In RL, where simulation environments play a bigger role, the robot agent actively explores the environment and receives information about which actions help fulfill the task through rewards. Next to adjusting its learning through hyperparameters, RL also requires a definition of rewards, depending on the task, and a description of the actions the robot can perform [36].

3.2. Monetizing Expertise: Application Expert Integration

Guided by the idea that a collaboration of humans and machines "is better than either humans or technology alone" [37], in multiple brainstorming sessions, we identified opportunities for application expert supervision to enhance the robot training and reduce the number of simulations to those that fit the later application context. We believe this context-focused approach can optimize training outcomes (like Prakash et al. [3]) without requiring prior assumptions about the context or engineering requirements. It could also represent a paradigm shift in how robot training is conceptualized and executed.

3.2.1. Validation and Improvements of Real-World Scans (Figure 1A)

In the current workflow (gray circle), virtual representations are constructed from environmental scans, mostly utilizing visual perception sensors for environment samplings [10]. However, these can exhibit shortcomings due to their type, the complexity of the scene, or the algorithms employed in the reconstruction process. The sensor's data quality can differ, depending on the sensor's resolution and precision, affecting the accuracy and reliability of the recordings [38]. For instance, scanning object surfaces with optical sensors can introduce artifacts, which are often a consequence of reflection, the distance at which the acquisition is made, and varying environmental conditions [38]. In addition, the complexity of scenes, especially those with occlusions, reflective surfaces, and transparent objects, necessitates acquiring data from multiple viewpoints to ensure comprehensive capturing of all elements. This requirement often leads to either the introduction of artifacts or the omission of critical information, necessitating corrective measures through the acquisition of data from varied angles. Active exploration of scenes typically leads to high computational and time costs since most approaches are based on sampling approaches and lack human intuition and real-world understanding. For instance, active vision approaches generally compute the best view by sampling all possible views and computing the scores for each view [39,40]. The higher the resolution of the scene and views, the higher the overall cost is, and these tend to increase exponentially. Learning-based approaches are often application-specific and require huge amounts of domain-specific data for training the models [41].

As application experts are familiar with the robot's operating environment [29,42], they can detect missing information and identify errors (e.g., missing objects) by reviewing the initial recordings; thereby, they can also assess scan quality. For instance, they may assist in filtering the recordings by selecting high-quality images or deleting faulty ones, thus improving the overall acquisition set. If insufficient images are found, they could initiate new scans or retakes. Furthermore, once application experts have identified an area with missing information, they could assist in pointing out areas of missing data, for example, by marking them, directly providing new sensor positions, or steering the capture via teleoperation for a complete capture. On-site AR systems may assist in this process by overlaying the scans with the original environment. Algorithmic approaches that suggest new scan positions can also be utilized during the interaction. These include algorithmic suggestions based on predictive models or real-time analysis, which experts can refine for optimal scan positions. Since steering a robot with human suggestions may be limited to the robot's operation area during acquisition (e.g., a static robotic arm that can only operate in a specific range), such approaches may require communication of the robot's characteristics to the application experts.

3.2.2. Correcting Virtual Representations (Figure 1B)

Creating virtual objects from scene scans involves various techniques, such as segmentation, filtering, classification, and mesh generation (described in Section 3.1.2). During segmentation, all data points belonging to an object must be identified and labeled. Segmentation algorithms based on traditional and machine learning (ML) algorithms can produce high-quality results [38] but also inaccurate and unreliable results in many cases. Noisy segmentations result in coarse boundaries between entities in the scenes, and in some cases, identify ghost objects that are not present in the scene. This is also due to algorithms relying on databases, which remain incomplete [43]. Once all object points have been identified, object meshes can be generated using mesh triangulation algorithms to enable interaction with virtual objects. However, computing meshes can be challenging [44] due to assumptions about the environment and the need for pre-defined parameters, such as requiring a constant point density in the raw images or planar surfaces in the output mesh [45]. Thus, they often yield noisy meshes that do not represent objects. Both inaccuracies from segmentation and mesh reconstruction necessitates uncertainty quantification when a highly reliable reconstruction is needed.

Humans excel at object recognition [38], which makes them valuable for improving the computed virtual representations of recordings using their scene understanding [29,42]. To enhance the recorded virtual scans, they could specify data points, extend planar surfaces, insert single or clusters of data, draw missing objects, or add objects from collections. Such intervention could also be performed iteratively with databases or generative algorithms (as in VRFromX [26]) to harness the respective strengths of humans and computers. Application experts could further contribute to the segmentation of objects from environmental scans such as point clouds by manually selecting objects, adjusting mesh boundaries, colors, and labels, or removing erroneous data [8,46] if a computed segmentation is incorrect or cannot be performed. The next stage involves creating meshes from the identified objects to enable interaction with virtual objects. Application experts could directly intervene by grasping and moving the boundaries of erroneous meshes. For instance, when working with point cloud data, these could be utilized as adjustment points to snap mesh boundaries during interaction. Another research opportunity is exploring how application experts could assist in creating textured meshes and improving the virtual representations of objects. To generate textures or adapt the visual representation, including a user-controlled generative artificial intelligence (AI) or databases for adaptation could further support the adaptation process.

3.2.3. Specifying Application-Specific Object Properties (Figure 1C)

Given an environment with interactable virtual objects, one challenge is reducing the number of generated simulation environments using DR to only those that are realistic and application-specific to improve training results. Real-world environments change through human intervention, as their habits, legal regulations, and work processes influence an object's state, which is expressed through its properties. Hence, robot agents may need to learn the physical and temporal properties of the objects and the scene in order to build a realistic simulation environment [47,48]. However, approaches for the agent to actively explore and understand such properties are unreliable and difficult to implement, and current works are at a very nascent stage. Thus, such approaches are often excluded [49] from robot training.

In such cases, human expertise can be crucial to incorporating human behavior in simulation environment generation. We envision the application experts either specifying their usage of objects in an environment directly or providing data about their behavior (e.g., with technologies such as Internet of Things (IoT) sensors). Application experts, with their domain expertise of the scenario and execution, could further specify parameters such as object positions, restrictions to their degrees of freedom (DoF), relations, occurrence probabilities, temporal characteristics, physical characteristics, and other properties. In addition, regulations, limitations, and users' preferences influence their object interactions. A structured process and interface for experts to input these parameters would be beneficial to obtain this information in a way that can be used for the simulation. Furthermore, visualization techniques in MR could assist by representing abstract concepts such as uncertainty or object relations. Hence, incorporating them could also lead to more context-specific selection of simulation environments. Including such human-based influences for building realistic scenarios could mean that visualizing abstract properties (e.g., uncertainty or object relations) could be challenging, despite them potentially aiding the specification of the spatial environment. Another task that could influence the training and is linked to the simulation is providing robot task information, considering subjective user preferences. For instance, depending on the application context, an application expert may prefer specific execution, such as sorting a fridge or a cupboard in a particular way or excluding areas from cleaning.

3.2.4. Verifying and Influencing Simulation Environment Generation (Figure 1D)

Utilizing DR leads to the generation of multiple simulation environments with varied randomized properties (Section 3.1.3). However, DR still suffers from intrinsic challenges. Besides the object properties to be randomized, it also requires algorithmic parameters for constrained randomization to receive application-specific simulations [3]. However, modeling such specific simulations requires time [3] and knowledge about the application context.

Since application experts are unfamiliar with RL or robotics, they may not directly contribute to parameters for DR. However, using their domain knowledge could assist in verifying whether the generated simulation environments represent real-world scenes that occur in the expected context. By incorporating application-specific object properties (see Section 3.2.3) and iterating on the subsequently generated environment (e.g., using previews), experts can ensure that only simulation environments matching their experiences are created. They can identify discrepancies and adjust object properties in the previously mentioned stages or delete the incorrect configurations to adjust the generation. In addition, they could help select diverse simulation environments that cover a range of real-world scenarios for training purposes and provide scenario probabilities to consider when determining their appearance frequencies during training. We see potential in evaluating how such feedback could influence training results.

3.2.5. Verifying Robot Training (Figure 1E)

Lastly, the robot agent learns to perform the task by training in the generated simulation environments. In addition to defining rewards and robot actions, RL requires hyperparameter tuning, which can be time-consuming [36] and results in the agent learning multiple strategies (policies) due to the myriad of generated environments. Ultimately, developers have to decide which policy to apply [18]. As non-machine learning experts, application experts may not have the expertise to select the optimal hyperparameters. However, they can assist in specifying rewards [50] since they understand the task requirements and the limitations of the process, as shown in Reinforcement Learning with Human Feedback (RLHF) [51]. Tasks can also be user-specific if there are preferences for execution, which could be considered in the training. They could observe the agent's performance in the simulation environment and assist in selecting the learned policy that best suits their use case (e.g., by watching a virtual robot perform actions using different policies and choosing the one that best suits their context). Hence, application experts could relieve machine learning developers of having to choose a policy. They could also indicate unwanted behavior or missing actions using methods for flagging and reporting these issues, as feedback on the learned behavior was shown to benefit the training [52].

4. Case Study

In the previous section, we described how application experts can use their domain knowledge to assist in the generation of application-specific simulation environments. In the following, we describe the initial insights of a case study in which we showcase several prototypes, each implementing an idea we previously proposed. Our primary goal is to demonstrate the feasibility of our proposed ideas, and our secondary goal is to gather insights into the different stages of the introduced workflow.

There are multiple technologies available through which application experts can be included in the robot training process, such as desktop solutions and MR technologies [25]. However, MR technologies, such as Augmented Reality (AR) and Virtual Reality (VR), offer great potential because they facilitate intuitive interaction with visual representations in three-dimensional space [8,29]. Thus, we choose to use VR to enable remote operation, allowing persons not physically present to interact with the system. We developed the following prototypes in VR using Unity3D and deployed the applications on the Meta Quest 2 as the selected VR head-mounted display.

4.1. Validation and Improvements of Real-World Scans

In our proposed workflow, we detailed opportunities for application experts to validate and enhance real-world scans by detecting missing information and initiating new scans (see Section 3.2.1). We concentrated on a scenario in which a robot has to perform a 3D scan of a scene as the first stage of transferring real-world knowledge into the simulation. We want to assess the degree to which a human expert can assist in this task. Multiple scans are needed to capture a three-dimensional scene, covering different perspectives. Therefore, we focused on the different sensor poses and implemented a VR environment in which users can define these poses, such that the robot can use them for the recording. We implemented two variations for interaction: one where users directly place the sensor positions and another where they mark areas with missing data, letting the system calculate scan positions from their input. Using the first variant (see Figure 2A), users can set and delete cameras inside the VR environment using a button press. Each camera displays its field of view to communicate its coverage when taking scans. Users can draw planes above areas with missing information with the second variant (see Figure 2B). Their size and angle are used to calculate the sensor position needed to cover the plane entirely. We prepared two scenes with different objects and positions to cover the different occlusions and complexities of objects. A user starts with one initial scan of the scene that is incomplete and adds camera positions using one of the interaction methods. After adding one or more poses, they can request that the remote robot take the images of the provided poses. Upon sampling, the point cloud representing the scanned scene is updated with the new information. Users can then choose to add new poses iteratively or confirm the representation as complete when satisfied. For the setup, we used the Panda robotic arm from Franka Emika (Franka Emika. https://www.franka.de/, last accessed: 20 November 2023), which was placed on a table and controlled using ROS1 and MoveIT. We utilized a depth sensor from RealSense, whose scans were used to calculate a mesh of the scene that was then transferred into a point cloud representation. All data was exchanged between the robot unit and the VR application in Collada format.

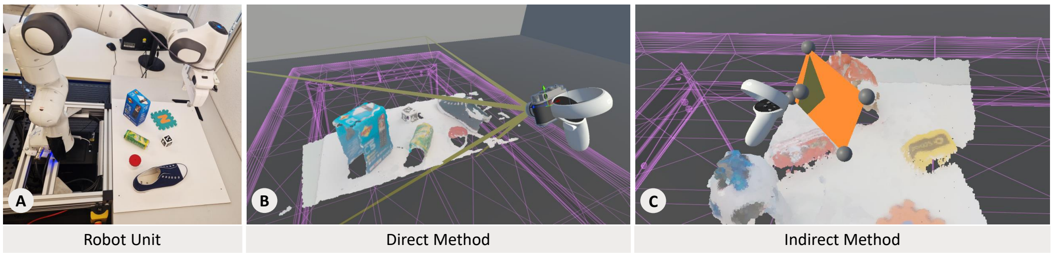

Figure 2. To direct an entire scene capture, we tested two interaction methods in a teleoperation setup with a robotic arm (**A**); Using the direct method, users can place and modify the cameras directly with the controllers (**B**); With the indirect method, they can specify missing areas in the scan using a virtual plane (**C**); The application then spawns a camera that covers the defined plane.

A preliminary self-exploration indicated that both methods have merits. The direct position input felt intuitive, as it was designed to mimic real-life photography, but the indirect method was beneficial when defining fine-grained areas of missing information, due to the indication of what should be recorded. Overall, we saw that remotely directing scan captures via a robotic arm is feasible using our approach. However, a comparison with automatic algorithms such as next-best view (NBV) shows that planning is needed to fully understand the efficiency and time efficiency differences between human and algorithm-based sampling.

4.2. Correcting Virtual Representations

We identified multiple opportunities for application experts to improve virtual representations obtained from environmental scans, including initial scan correction, segmentation, mesh generation, and addition of missing scene objects (see Section 3.2.2). Since we recognized that the segmentation and mesh generation are often error-prone, we tested ways to involve application experts in segmenting point cloud scans obtained from real-world scans using two different implementations.

Using the natural perspective within the 3-dimensional environment of VR, we first aimed to create an easy-to-use segmentation tool. We started our initial prototype by implementing a bounding box for the segmentation task, spanning between both controllers (see Figure 3A). It selects or deselects points within it based on the user's mode choice, thus functioning as a universal segmentation tool. To segment an object, users start in the VR application to view the initial environmental scan. They can select or deselect points using the selection tool and, upon confirmation, delete them. This process allows for iterative refinement, as the user removes points not part of the target object until it is fully segmented (see Figure 3A, right). To test if the implementation is usable both for simple and complex scene constellations, we included an example of each in our testing session.

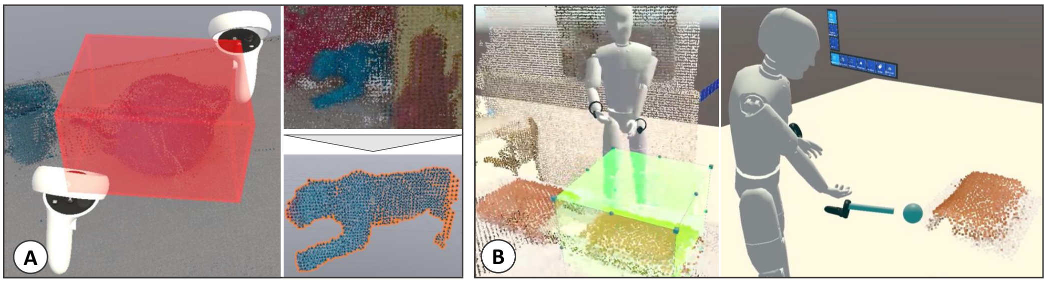

Figure 3. We enabled the segmentation of objects from point cloud scans using a box spanning between the controllers, such that users could segment even occluded objects inside a scene (**A**), and a coarse and a fine segmentation tool, realized through a bounding box enabling remote interaction and a sphere attached to a stick (**B**).

Our self-assessment of the prototype revealed that the segmentation tool was usable for both simple and complex scenes, though it required further refinement for more complex tasks. We found the natural view within VR beneficial during segmentation, especially for distinguishing between overlapping objects. In later work, we contrasted the implementation with comparable applications for desktop and tablet to assess which of these devices are most suitable for segmenting simple and complex point clouds regarding efficiency and effectiveness [8].

Next, we aimed to further enhance the segmentation in VR. We wanted to enable segmentation with different segmentation tools, as different characteristics may suit different tasks. Since we had only included a box in the previous application, we wanted to include a sphere-like segmentation tool for curved surfaces. Thus, we developed a coarse and a fine segmentation tool (see Figure 3B) using the Mixed Reality Toolkit (Microsoft. https://docs.microsoft.com/de-de/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05 (last visited on 20 November 2023)) (MRTK). For a coarse segmentation tool, we implemented a virtual box (similar to the previous application) that selects points within it and can be adjusted using the interactive markers via grasp or ray cast interaction. The fine tool, a sphere on a stick, deletes specific points and can be shrunken or enlarged. Post-segmentation, our system uses the Ball-Pivoting Algorithm [53] to generate object meshes, which can then be labeled. For testing, we downloaded six point clouds of objects and placed them in a three-dimensional scan of a bedroom [54] to assess segmentation quality. We sequentially segmented the six objects in a testing session using both segmentation tools.

Our initial tests revealed effective object isolation, with a segmentation accuracy of 96.7% in the median when comparing the object's point clouds to the segmentation result. However, switching between the segmentation tools and repositioning oneself during segmentation to obtain a good work position for the changes lead to higher segmentation times. Handling the fine segmentation tool was perceived as easy, but using ray casts to adjust the coarse segmentation was perceived as challenging.

4.3. Specifying Application-Specific Object Properties

Human behavior influences the object positioning based on the object usage (Section 3.2.3). To model how objects change due to interaction, application experts can specify the properties of the previously created virtual objects. Many object properties need to be modeled to generate realistic simulation environments, such as object weight, friction, overlap, transparency, and lighting. In addition, all need a pose, including translation properties (along x-, y-, and z-axes) and rotations (pitch, yaw, and roll). Furthermore, these poses could have different occurrence probabilities, which must be considered. Thus, we focused on object probabilities, including poses and relationships, for initial testing (see Figure 4). An object's pose, translation, and rotation can be specified just by moving it from the catalog into the VR scene. In addition to independent object poses, we specified parent-child relationships regarding object poses. When an object is specified as a child of another object, its position is considered in relation to its parent during an arrangement. This can lead to a translation or rotation in accordance with the parent object. For instance, a parent-child relationship can ensure that a pillow (child) is always on a bed or sofa (parent). In addition, we introduced a virtual slider for users to assign probabilities to these specifications. For testing, we modeled three scenarios using the application: a child's bedroom, a student's single-room apartment, and a living room.

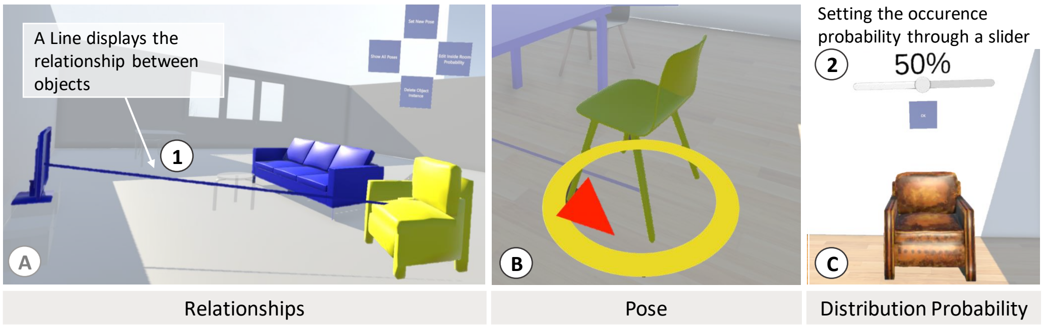
A Line displays the relationship between objects
1
A
B
Setting the occurence probability through a slider
2
50%
C
Relationships
Pose
Distribution Probability

Figure 4. We implemented the prototypes for application experts to specify (**A**) object relationships such as parent-child or sibling relationships using color highlighting and (1) connected lines for visualization, (**B**) object translations, object rotations, and arrows to specify an object's front side, as well as (**C**) object distribution probabilities, provided through (2) a slider to define how often it should appear in the generated simulations.

Using the system, we were able to model the different scenarios. The color highlighting was found to be helpful in obtaining an overview of the already-made specifications and viewing the inserted relationships. Working in the virtual environment was found to be satisfactory, although the number of specifications was time-consuming.

4.4. Verifying and Influencing Simulation Environment Generation

As detailed in Section 3.2.1, application experts can also assist in verifying the realism of the created simulation environments. Upon recognizing mismatches, they could point them out or iteratively adjust their specifications. With this in mind, we incorporated an animation of simulation environments generated from the specifications using the probability prototype for verification purposes. Hence, we customized our DR algorithm to incorporate the provided constraints (poses, relations, and probabilities) for placing the virtual objects. Inside the VR application, users can view different generated environments based on their specifications and skip between them using a button press (see Figure 5). Afterward, they can refine their inputs to fine-tune the randomization. Like before, we modeled three scenarios using the application for testing, enabling the new functionality of viewing the generated room instances: a child's bedroom, a student's single-room apartment, and a living room. The test person could enter the viewing mode at any time during the process but only complete a room after viewing the final outcome.

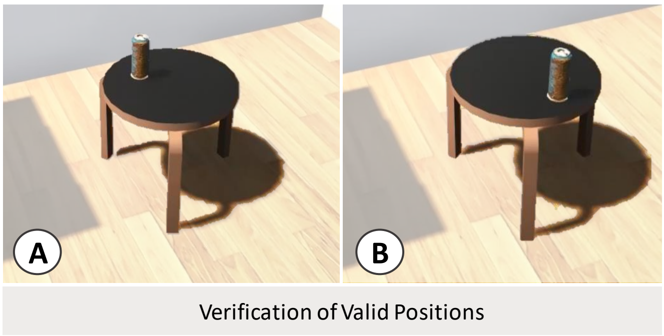

Figure 5. We enhanced our object probability prototype to include animations of created environments using the specified object properties to facilitate verification. It enabled a steerable sequence to view different object properties, such as varying can poses in the simulations, positioned either (**A**) farther or (**B**) closer on the table.

Compared to the version without the verification, the tester reported a higher level of certainty and an improved overview of the specifications. Furthermore, it led to faster recognition of errors, as they were directly visible in the generated scenes.

4.5. Outlook

In our case study, we showcased the feasibility of involving application experts in the simulation generation process for sim2real transfer, suggesting an enhancement in the quality and relevance of robotic applications in the different stages of our workflow is possible. It allows the integration of specialized knowledge, uniquely possessed by the system's end users. Despite having tested only a subset of our proposed ideas, we believe, in line with Schmidt et al. [37], that the synergy between human and machine expertise surpasses the capabilities of either humans or machines alone. Nevertheless, a comprehensive evaluation of the workflow is still needed, as is a comparison of the individual algorithms employed in its various stages. This opens a multitude of research opportunities. Future studies could shed light on the effectiveness of the collaboration, focusing on training outcomes, time efficiency, and its impact across various application contexts. Such investigations may enhance our understanding of the interplay between human expertise and machine intelligence in the realm of robotic applications.

5. Conclusions

This paper introduced a new conceptual workflow for human-in-the-loop sim2real transfer to utilize the application expert's domain knowledge for robot training. Based on existing research on robot training using sim2real, we derived a workflow for simulation-based robot training. We extended this workflow by outlining five main stages in which application experts can contribute to the generation of real-world simulation environments: (1) validating and improving real-world scans, (2) correcting virtual representations, (3) specifying application-specific object properties, (4) verifying and influencing the simulation environment generation, and (5) verifying the robot training. We highlighted research opportunities in each identified stage and explained how our human-in-the-loop approach can enhance robot training. Thereafter, we presented a case study in which we implemented different prototypes, demonstrating the potential of human experts in each of the five stages. We used VR as the interaction technology because of its three-dimensional rendering and intuitive head pose tracking. We expect that application experts can improve robot training by applying their context-specific knowledge to the training process. We hypothesized that this may reduce the complexity of the training process, making it more focused on the application at hand. However, a detailed evaluation of the workflow and comparison with individual algorithms used in different stages is yet to be conducted. Our early insights are promising and show that, despite strong efforts toward full automation, humans can offer valuable input that should not be underestimated or minimized in future work.

Author Contributions: Conceptualization, T.C.S., U.G. and S.S.; Methodology, T.C.S.; Software, C.L.; Validation, C.L.; Resources, C.L.; Writing—Original Draft Preparation, C.L.; Writing—Review & Editing, C.L., P.M., J.A., T.C.S., U.G. and S.S.; Visualization, C.L.; Project Administration, T.C.S. and S.S.; Funding Acquisition, M.P., T.C.S., U.G. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the German Federal Ministry of Education and Research (01IS21068B).

Institutional Review Board Statement: For our reported tests, we followed our institutional ethical guidelines. Our local ethics committee of the University of Duisburg-Essen permitted them.

Informed Consent Statement: Informed consent was obtained from all subjects involved.

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADR	Automatic Domain Randomization
AI	Artificial Intelligence
DoF	Degrees of Freedom
DR	Domain Randomization
HRI	Human-Robotics Interaction
IoT	Internet of Things
MDPI	Multidisciplinary Digital Publishing Institute
ML	Machine Learning
MR	Mixed Reality
NBV	Next-best view
RL	Reinforcement Learning
RLHF	Reinforcement Learning with Human Feedback
sim2real	Simulation-to-Real
VR	Virtual Reality

References

1. Leibovich, G.; Jacob, G.; Endrawis, S.; Novik, G.; Tamar, A. Validate on Sim, Detect on Real - Model Selection for Domain Randomization. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 7528–7535. [CrossRef]
2. Tobin, J.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30. [CrossRef]
3. Prakash, A.; Boochoon, S.; Brophy, M.; Acuna, D.; Cameracci, E.; State, G.; Shapira, O.; Birchfield, S. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 7249–7255. [CrossRef]
4. Ramos, F.; Possas, R.C.; Fox, D. BayesSim: Adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv* **2019**, arXiv:1906.01728.
5. Evans, B.; Thankaraj, A.; Pinto, L. Context is Everything: Implicit Identification for Dynamics Adaptation. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 2642–2648. [CrossRef]
6. Wirth, F.; Quchl, J.; Ota, J.; Stiller, C. PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1693–1698. [CrossRef]

7. Stets, J.D.; Sun, Y.; Corning, W.; Greenwald, S.W. Visualization and labeling of point clouds in virtual reality. In Proceedings of the SIGGRAPH Asia 2017 Posters, Bangkok, Thailand, 27–30 November 2017; pp. 1–2. [CrossRef]
8. Liebers, C.; Prochazka, M.; Pfützenreuter, N.; Liebers, J.; Auda, J.; Gruenefeld, U.; Schneegass, S. Pointing It Out! Comparing Manual Segmentation of 3D Point Clouds between Desktop, Tablet, and Virtual Reality. *Int. J. Hum. Comput. Interact.* **2023**, 1–15. [CrossRef]
9. Valentin, J.; Vineet, V.; Cheng, M.M.; Kim, D.; Shotton, J.; Kohli, P.; Nießner, M.; Criminisi, A.; Izadi, S.; Torr, P. SemanticPaint: Interactive 3D Labeling and Learning at Your Fingertips. *ACM Trans. Graph.* **2015**, *34*, 1–17. [CrossRef]
10. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [CrossRef]
11. Thoma, M. A Survey of Semantic Segmentation. *arXiv* **2016**, arXiv:1602.06541. [CrossRef]
12. Oleynikova, H.; Taylor, Z.; Fehr, M.; Nieto, J.I.; Siegwart, R. Voxblox: Building 3D Signed Distance Fields for Planning. *arXiv* **2016**, arXiv:1611.03631. [CrossRef]
13. Chebotar, Y.; Handa, A.; Makoviichuk, V.; Macklin, M.; Issac, J.; Ratliff, N.; Fox, D. Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8973–8979. [CrossRef]
14. OpenAI; Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; et al. Solving Rubik's Cube with a Robot Hand. *arXiv* **2019**, arXiv:1910.07113.
15. Mehta, B.; Diaz, M.; Golemo, F.; Pal, C.J.; Paull, L. Active Domain Randomization. In Proceedings of the Conference on Robot Learning, Online, 16–18 November 2020; Volume 100, pp. 1162–1176.
16. Muratore, F.; Ramos, F.; Turk, G.; Yu, W.; Gienger, M.; Peters, J. Robot Learning From Randomized Simulations: A Review. *Front. Robot. AI* **2022**, *9*, 799893. [CrossRef] [PubMed]
17. Beck, J.; Vuorio, R.; Liu, E.Z.; Xiong, Z.; Zintgraf, L.; Finn, C.; Whiteson, S. A Survey of Meta-Reinforcement Learning. *arXiv* **2023**, arXiv:2301.08028. [CrossRef]
18. Andrychowicz, O.M.; Baker, B.; Chociej, M.; Józefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **2020**, *39*, 3–20. [CrossRef]
19. Gu, J.; Xiang, F.; Li, X.; Ling, Z.; Liu, X.; Mu, T.; Tang, Y.; Tao, S.; Wei, X.; Yao, Y.; et al. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. *arXiv* **2023**, arXiv:2302.04659. [CrossRef]
20. James, S.; Ma, Z.; Arrojo, D.R.; Davison, A.J. RLBench: The Robot Learning Benchmark & Learning Environment. *arXiv* **2019**, arXiv:1909.12271. [CrossRef]
21. Dalal, M.; Mandlekar, A.; Garrett, C.; Handa, A.; Salakhutdinov, R.; Fox, D. Imitating Task and Motion Planning with Visuomotor Transformers. *arXiv* **2023**, arXiv:2305.16309. [CrossRef]
22. Mandlekar, A.; Nasiriany, S.; Wen, B.; Akinola, I.; Narang, Y.; Fan, L.; Zhu, Y.; Fox, D. MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. In Proceedings of the 7th Annual Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023.
23. Montano-Murillo, R.A.; Nguyen, C.; Kazi, R.H.; Subramanian, S.; DiVerdi, S.; Martinez-Plasencia, D. Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments. In Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Atlanta, GA, USA, 22–26 March 2020; pp. 53–62. [CrossRef]
24. Blanc, T.; Verdier, H.; Regnier, L.; Planchon, G.; Guérinot, C.; El Beheiry, M.; Masson, J.B.; Hajj, B. Towards Human in the Loop Analysis of Complex Point Clouds: Advanced Visualizations, Quantifications, and Communication Features in Virtual Reality. *Front. Bioinform.* **2021**, *1*, 775379. [CrossRef] [PubMed]
25. Wang, Z.; Nguyen, C.; Asente, P.; Dorsey, J. DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama Japan, 8–13 May 2021; pp. 1–12. [CrossRef]
26. Ipsita, A.; Li, H.; Duan, R.; Cao, Y.; Chidambaram, S.; Liu, M.; Ramani, K. VRFromX: From Scanned Reality to Interactive Virtual Experience with Human-in-the-Loop. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama Japan, 8–13 May 2021; pp. 1–7. [CrossRef]
27. Chakraborti, T.; Sreedharan, S.; Kulkarni, A.; Kambhampati, S. Projection-Aware Task Planning and Execution for Human-in-the-Loop Operation of Robots in a Mixed-Reality Workspace. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4476–4482. [CrossRef]
28. Gancet, J.; Weiss, P.; Antonelli, G.; Pfingsthorn, M.F.; Calinon, S.; Turetta, A.; Walen, C.; Urbina, D.; Govindaraj, S.; Letier, P.; et al. Dexterous Undersea Interventions with Far Distance Onshore Supervision: The DexROV Project. *IFAC PapersOnLine* **2016**, *49*, 414–419. [CrossRef]
29. Krings, S.C.; Yigitbas, E.; Biermeier, K.; Engels, G. Design and Evaluation of AR-Assisted End-User Robot Path Planning Strategies. In Proceedings of the 2022 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Sophia Antipolis, France, 21–24 June 2022; pp. 14–18. [CrossRef]
30. Metzner, M.; Bönig, J.; Blank, A.; Schäffer, E.; Franke, J. "Human-In-The-Loop"-Virtual Commissioning of Human-Robot Collaboration Systems. In *Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 131–138. [CrossRef]

31. Xia, F.; Shen, W.B.; Li, C.; Kasimbeg, P.; Tchapmi, M.E.; Toshev, A.; Martin-Martin, R.; Savarese, S. Interactive Gibson Benchmark: A Benchmark for Interactive Navigation in Cluttered Environments. *IEEE Robot. Autom. Lett.* **2020**, *5*, 713–720. [CrossRef]
32. Vuong, Q.; Vikram, S.; Su, H.; Gao, S.; Christensen, H.I. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies? *arXiv* **2019**, arXiv:1903.11774. [CrossRef]
33. Han, M.; Zhang, Z.; Jiao, Z.; Xie, X.; Zhu, Y.; Zhu, S.C.; Liu, H. Reconstructing Interactive 3D Scenes by Panoptic Mapping and CAD Model Alignments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 12199–12206. [CrossRef]
34. Kormushev, P.; Calinon, S.; Caldwell, D. Reinforcement Learning in Robotics: Applications and Real-World Challenges. *Robotics* **2013**, *2*, 122–148. [CrossRef]
35. Xia, F.; Zamir, A.R.; He, Z.; Sax, A.; Malik, J.; Savarese, S. Gibson Env: Real-World Perception for Embodied Agents. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 9068–9079.
36. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
37. Schmidt, A.; Herrmann, T. Intervention User Interfaces: A New Interaction Paradigm for Automated Systems. *Interactions* **2017**, *24*, 40–45. [CrossRef]
38. Barnefske, E.; Sternberg, H. Evaluating the Quality of Semantic Segmented 3D Point Clouds. *Remote Sens.* **2022**, *14*, 446. [CrossRef]
39. Batinovic, A.; Petrovic, T.; Ivanovic, A.; Petric, F.; Bogdan, S. A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration. *arXiv* **2020**, arXiv:2011.02182. [CrossRef]
40. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding Horizon "Next-Best-View" Planner for 3D Exploration. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1462–1468. [CrossRef]
41. Mendoza, M.; Vasquez-Gomez, J.I.; Taud, H.; Sucar, L.E.; Reta, C. Supervised Learning of the Next-Best-View for 3D Object Reconstruction. *arXiv* **2019**, arXiv:1905.05833. [CrossRef]
42. Yigitbas, E.; Sauer, S.; Engels, G. Using Augmented Reality for Enhancing Planning and Measurements in the Scaffolding Business. In Proceedings of the Companion of the 2021 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Online, 8–11 June 2021; pp. 32–37. [CrossRef]
43. Liu, Y.; Zulfikar, I.E.; Luiten, J.; Dave, A.; Ramanan, D.; Leibe, B.; Ošep, A.; Leal-Taixé, L. Opening up Open-World Tracking. arXiv **2022**, arXiv:2104.11221. [CrossRef]
44. Chen, K.; Yin, F.; Wu, B.; Du, B.; Nguyen, T. Mesh Completion with Virtual Scans. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AL, USA, 19–22 September 2021; pp. 3303–3307. [CrossRef]
45. Sorensen, T.A.; Mark, N.; Mogelmose, A. A RANSAC Based CAD Mesh Reconstruction Method Using Point Clustering for Mesh Connectivity. In Proceedings of the 2021 International Conference on Machine Vision and Applications, Nagoya, Japan, 25–27 July 2021; pp. 61–65. [CrossRef]
46. Jackson, B.; Jelke, B.; Brown, G. Yea Big, Yea High: A 3D User Interface for Surface Selection by Progressive Refinement in Virtual Environments. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Online, 18–22 March 2018; pp. 320–326. [CrossRef]
47. Jiang, H.; Mao, Y.; Savva, M.; Chang, A.X. OPD: Single-view 3D Openable Part Detection. *arXiv* **2022**, arXiv:2203.16421. [CrossRef]
48. Sturm, J.; Stachniss, C.; Burgard, W. A Probabilistic Framework for Learning Kinematic Models of Articulated Objects. *arXiv* **2014**, arXiv:1405.7705. [CrossRef]
49. Zaidenberg, S.; Reignier, P.; Mandran, N. Learning User Preferences in Ubiquitous Systems: A User Study and a Reinforcement Learning Approach. In *Artificial Intelligence Applications and Innovations*; Papadopoulos, H., Andreou, A.S., Bramer, M., Eds.; IFIP Advances in Information and Communication Technology Series; Springer: Berlin/Heidelberg, Germany, 2010; Volume 339, pp. 336–343. [CrossRef]
50. Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; Amodei, D. Reward learning from human preferences and demonstrations in Atari. *arXiv* **2018**, arXiv:1811.06521. [CrossRef]
51. Li, G.; Gomez, R.; Nakamura, K.; He, B. Human-Centered Reinforcement Learning: A Survey. *IEEE Trans. Hum.-Mach. Syst.* **2019**, *49*, 337–349. [CrossRef]
52. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* **2022**, arXiv:1811.06521. [CrossRef]
53. Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Trans. Vis. Comput. Graph.* **1999**, *5*, 349–359. [CrossRef]
54. Caine, M. B&B—Sitescape: 3D Model. 2021. Available online: https://sketchfab.com/3d-models/bb-sitescape-0eb74dd92a534a37880a6d2b3aa980b5 (accessed on 20 November 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.