2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR) | 979-8-3503-7202-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIxVR59861.2024.00025

CIS2VR: CNN-based Indoor Scan to VR Environment Authoring Framework

Hiranya Kumar
The University of Texas at Dallas
hiranya@utdallas.edu

Ninad Khargonkar
The University of Texas at Dallas
ninadarun.khargonkar@utdallas.edu

Balakrishnan Prabhakaran
The University of Texas at Dallas
bprabhakaran@utdallas.edu

***Abstract*—We present CIS2VR (CNN-based Indoor Scan to VR), an authoring framework designed to transform input RGB-D scans captured by conventional sensors into an interactive VR environment. Existing state-of-the-art 3D instance segmentation algorithms are employed to extract object instances from RGB-D scans. A novel 3D Convolutional Neural Network (3D CNN) architecture is used to learn 3D shape features common to both classification and 3D pose estimation problems, enabling rapid shape encoding and pose estimation of objects detected in the scan. The generated embedding vector and predicted pose are then used to retrieve and align a matching 3D CAD (Computer-Aided-Design) model. The aligned models, along with the estimated layout of the scene, are transferred to Unity, a 3D game engine, to create a VR scene. An optional human-in-the-loop system allows users to validate results at various steps of the pipeline, improving the quality of the final VR scene. We evaluate and compare our approach to existing semantic reconstruction methods on key metrics. The proposed approach outperforms several existing methods in object alignment, coming close to the state-of-the-art, while speeding up the process an order of magnitude. CIS2VR takes an average of 0.68 seconds for the entire conversion across our test dataset of 312 scenes. The code for the proposed framework will be made publicly available on GitHub.**

***Index Terms*—Artificial Intelligence, Virtual Reality, Deep Learning, Machine Learning**

I. Introduction

Interest in Virtual, Augmented, and Mixed Reality (VR, AR, MR: together referred to as Extended Reality/XR systems) systems has grown significantly in recent years owing to the widespread adoption of commodity depth-sensing devices (Microsoft Kinect, iPhones, etc.) and improvements in Human Interface Devices (HID) for XR systems such as Microsoft Hololense 2, Apple Vision Pro, Meta Quest 3, etc. However, the accessibility of XR experiences has been restricted, partially attributable to the resource-intensive and time-consuming process inherent in the development of 3D experiences and related 3D assets. The replication of real-world scenes for XR systems further compounds the intricacies of the development procedure. Frameworks facilitating the automated conversion of RGB-D data into a 3D environment hold substantial promise in alleviating these development challenges.

3D reconstruction techniques can be broadly classified into two types based on their output: dense 3D reconstruction and 3D semantic reconstruction. A substantial body of research has been devoted to dense 3D reconstructions [1]- [13]. Pioneering efforts like KinectFusion [1] and StereoScan [11] introduced algorithms capable of generating dense, precise, and smooth 3D surface reconstructions from real-time RGB-D videos. Subsequent advancements, exemplified by works such as ElasticFusion [13] and BundleFusion [15], have further refined various facets of the reconstruction process. While 3D environments generated through such methods can boast detailed and photorealistic attributes, they inherently manifest noise and incompleteness due to factors like scanning patterns, camera viewpoints, and sensor noise. Furthermore, these environments often entail substantial file sizes (ranging from hundreds of megabytes to multiple gigabytes depending on point density) with the major drawback that *individual objects in the scene cannot be interacted with*, which makes them suboptimal for collaborative or interactive applications.

3D Semantic reconstruction techniques [14]- [18] address these challenges by substituting objects in the scene with CAD (Computer Aided Design) models that are semantically and geometrically similar to objects in the RGB-D scan. While scenes generated through 3D semantic reconstruction may not achieve the same level of photorealism or geometric precision as dense 3D reconstructions, they offer completeness, reduced file sizes, and interactivity. The representation of scene objects using individual CAD models enables the resulting 3D environment to incorporate details such as texture, material composition, and other physical characteristics of each object. This proves crucial for VR experiences and 3D simulations where the physical attributes of objects significantly contribute to the overall experience. Semantic 3D reconstruction presents several challenges, primarily along with object detection within the scene and the subsequent retrieval and alignment of CAD models corresponding to each identified object. Recent contributions in semantic reconstruction [14] [15] [16] [17] [18] have made notable progress in confronting these challenges. However, to the best of our knowledge, there is currently no existing work that comprehensively benchmarks and transforms an input RGB-D scan into a functional 3D VR scene.

Proposed Approach: We present an authoring framework CIS2VR (CNN-based Indoor Scan to VR), an authoring framework designed to transform an input indoor RGB-D scene into a functional 3D VR environment in the Unity framework. We start with a state-of-the-art 3D instance segmentation algorithm for detecting object instances in the input RGB-D

2771-7453/24/$31.00 ©2024 IEEE
DOI 10.1109/AIxVR59861.2024.00025

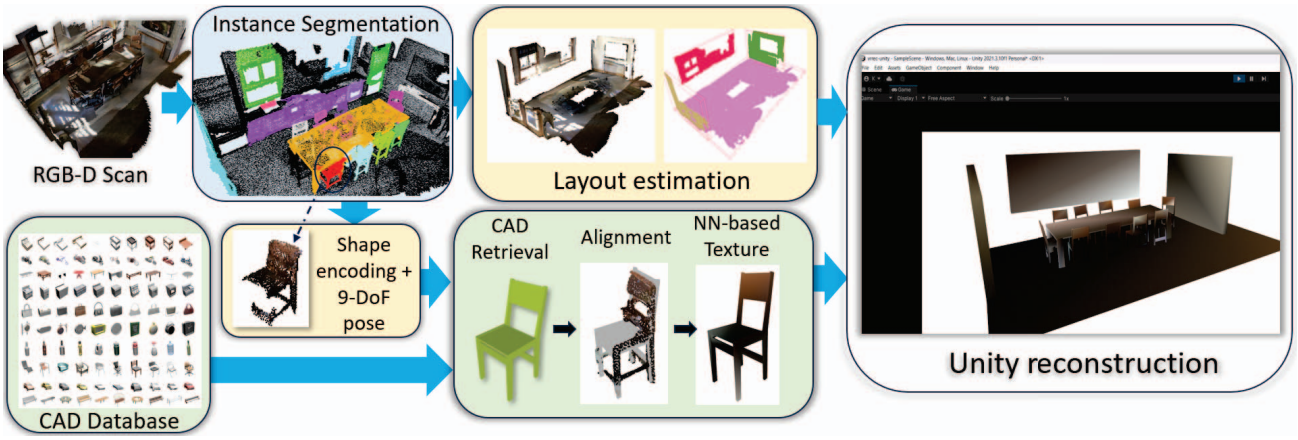

Fig. 1: An overview of CIS2VR (CNN-based Indoor Scan to VR).

scan. Detected instances are then processed with a novel 3D Convolutional Neural Network (CNN), inspired by Choy et. al [19], that learns feature space common to both scan and CAD objects and generates a shape encoding for 9 Degrees of Freedom (DoF) pose (translation, rotation, and scale, each along x, y, z axes). The resultant shape encoding is used to retrieve a corresponding CAD model from the database, with the predicted pose facilitating the alignment of the CAD model to the object instance. The aligned models are transferred to a Unity scene for downstream use cases.

Contributions: CIS2VR is an end-to-end framework that transforms an indoor RGB-D scan into a functional 3D VR scene with the following salient features.

- A novel and efficient 3D CNN architecture that concurrently learns a shared feature space for both scan and CAD objects while also predicting their category and 9-DoF pose simultaneously in a single pass.
- CIS2VR serves as an authoring framework to create a semantically and visually similar VR scene in Unity from an RGB-D scan.
- We incorporate a flexible human-in-the-loop approach to allow users to modify the results of the system.
- The proposed method achieves close to the state-of-the-art alignment performance while demonstrating significantly reduced runtime. CIS2VR takes an average of 0.68 seconds for the entire conversion across our test dataset of 312 scenes.
- The code for the proposed framework will be made publicly available on GitHub.

II. Related Work

There exists a rich line of research on 3D reconstruction from diverse inputs such as multi-view RGB images, fused RGB-D scans, and point clouds. Prior works have also looked at reconstructing entire scenes in an online fashion and real-time constraints, as seen in KinectFusion [1] [25], BundleFusion [2] and NeuralRecon [26]. However, most of them focus on producing a mesh or TSDF output with predefined quality constraints and rely on color information for accurate representation. The reconstruction based on CAD models facilitates enhanced user interaction with the environment and grants greater freedom to VR environment designers, enabling them to manipulate and alter object parameters without necessitating post-processing on noisy reconstructed meshes.

A. CAD Model Retrieval and Alignment

Machine learning approaches for aligning CAD models to 3D scenes have been studied both from classical, hand-tuned features [29] [30] [31] perspective and deep learning-based methods [17]. Song et al. [29] use linear SVMs (Support Vector Machines) corresponding to each model in a CAD dataset and iteratively go over the scene via a sliding window approach. Li et al. [28] obtain shape descriptors and key points for both noisy scans and 3D CAD models, encoding their geometric features for efficient matching. Gupta et al. [17] leverage CNNs to output probable poses for objects already segmented by an existing segmentation method. ASIST [31] tries to have a single, unified pipeline for semantic labeling and retrieval via an energy-based formulation for an input point cloud. 3DMatch [35] developed a Siamese neural network as a feature extractor for establishing correspondences between the input scans and models.

Scan2CAD [14] proposes a 3D CNN approach to target similar correspondences and address issues with the domain gap of real-world scans and CAD models. They also introduce an annotated dataset for CAD model retrieval and alignment based on ScanNet [36] and ShapeNet [37]. Dahnert et al. [16] proposed a joint embedding space between CAD models and scanned objects based on existing Scan2CAD model annotated datasets. SceneCAD [15] considers the layout reconstruction part of the overall problem where instead of independently assessing each object, a graph neural network is used to enforce consistency in the scene's reconstruction. Other approaches have used a single RGB image of a scene to retrieve and align CAD models. Lim et al. [42] use an RGB image to estimate the relative pose between a CAD model and an object in the image. Huang et al. [44], Manni et al. [39], and

Izadinia et al. [43] use an RGB image to perform semantic 3D reconstruction, relying on depth data inferred from deep learning models.

B. Virtual World Reconstruction

Some of the algorithmic ideas have been primarily used in applications pertaining to Virtual, Augmented and Mixed Reality (VR/AR/MR) where creating an environment from scratch might be time-consuming. RealitySkins [38] tries to address this problem by dynamically generating the environment based on the input scan from a user's head-mounted display (HMD). Snap2Cad [39] shows a system that utilizes the built-in RGB sensors on modern smartphones to reconstruct an object in AR via matching with CAD models for use in online multiplayer scenarios. VRFromX [40] also uses neural network-based methods for object retrieval and alignment with human-in-the-loop as a part of an interactive content creation tool. In a similar vein, TransforMR [41] is a mixed reality system for object substitution that leverages segmentation and accurate pose estimation for consistent replacement and use in character animation.

III. CIS2VR Design

CIS2VR converts an input indoor RGB-D scan into a semantically and visually similar, interacTable IIID scene in Unity, as described in Figure 1. To achieve our goal, we start by segmenting objects in the input RGB-D scene utilizing an instance segmentation algorithm introduced by Vu et al. [21]. Subsequently, we process the extracted object instances through our 3D shape encoder and 9-DoF (translation, rotation, and scale, each along x, y, z axes) pose estimation algorithm, which generates a vector embedding for retrieving a geometrically matching CAD model and predicts the pose for aligning the matched CAD model to the RGB-D object instance in one forward pass. We use a K-Nearest Neighbors (KNN) based search algorithm in conjunction with the predicted semantic label to fetch a matching model from our CAD model database built using the ShapeNetV2 dataset [37]. To reconstruct the structural components of the scene, such as walls, floor, and ceiling, we isolate points corresponding to the relevant classes from the instance segmentation results. We employ RANSAC to further filter out points associated with each structure, subsequently using a 3D Oriented Bounding Box (OBB) to quantify the alignment of the structures. The information on retrieved CAD models and their corresponding poses is passed to Unity, which then retrieves and aligns each model, effectively reconstructing the scene. The subsequent sub-sections provide further details on each step of the process.

A. Input Data

RGB-D is a widely used input modality to capture 3D data, with high-quality annotated indoor RGB-D datasets such as S3DIS [46], ScanNet [36], and SUN-RGB-D [47] available today. Two types of RGB-D data are widely available: RGB-D images and RGB-D scans. RGB-D images resemble a 2D RGB image augmented with depth data derived from a depth sensor. Consequently, they may exhibit blind spots, signifying areas with missing depth data due to occlusion from various objects in the scene. Furthermore, RGB-D images provide a restricted, single-perspective view of the scene. This limitation is suboptimal for reconstructing entire scenes, as objects with missing parts are susceptible to misidentification, and the confined field of view captures only a fraction of the actual scene. RGB-D scans overcome this issue by combining several RGB-D images, either from an RGB-D video or individual RGB-D images taken from different viewpoints to cover for blind spots. Consequently, RGB-D scans boast superior point density, capturing finer details, maintaining consistent completeness in object depiction, and encompassing more, if not the entirety, of the indoor scene in a single scan. Due to these advantages, we opt to utilize RGB-D scans as the input for the proposed system. For this implementation, we use scenes from ScanNet v2 dataset [36] for training and evaluation.

B. Object Detection and Segmentation

To semantically reconstruct a given 3D scene, the initial step involves the detection and extraction of objects present in the input RGB-D scan. Owing to breakthroughs in 3D deep learning architectures and advancements in computational hardware making training large models more accessible, CNN-based 3D object detection and segmentation methods have made significant strides [21] [22] [48]. Recent contributions in this domain [20] [21] [32] [48] [49] can be broadly categorized into bounding-box-based methods (object detection algorithms) [21] [19] [32] [48] and mask-based methods (semantic segmentation algorithms) [20] [21] [49]. Bounding box-based methods output a 3D bounding box for each object instance in the scene, while mask-based methods produce semantic and instance labels for each point in the input point cloud. In indoor scenes, objects are often in close proximity, posing a challenge for bounding box-based methods as they can be imprecise in extracting points corresponding to a specific instance in the presence of overlap. This overlap may cause the bounding box to encompass points from adjacent objects, potentially altering the geometric characteristics of the instance and misleading the CAD model retrieval algorithm. To circumvent this issue, we opt for a mask-based method that generates precise segmentation masks for each instance in the scene. Each point P_i in the scene point cloud is associated with a singular instance and semantic label in a segmentation mask, eliminating the potential for overlap between segmented point clouds of different object instances and resolving the aforementioned problem.

C. Model Retrieval and Alignment

With the goal of generating an interactive VR scene, the imperative is to replace objects in the scene with aligned 3D object models that possess both semantic and geometric similarity. Previous works have explored different ways to address this challenge. Some early approaches have used template matching with hand-crafted per-class templates [50]. More recent works have utilized large model datasets like

ShapeNet [37] and ModelNet [51] in conjunction with 3D CNNs for their model retrieval tasks [14] [16], using the CNN to generate 3D shape encoding. Matching models are retrieved by using a nearest-neighbor-based search. This ensures that the system can handle a very large number and wide variety of models for each semantic class while keeping the run-time computational costs low.

1) CAD Retrieval and 9-DoF Pose: Object classification networks are adept at learning features for distinguishing between objects with high accuracy. We posit that these features, designed for object classification, could also be advantageous for object pose estimation. To harness these features for 3D pose estimation and allow the joint prediction of object category and pose, we propose a novel 3D CNN architecture inspired by [19]. This architecture is designed to concurrently learn semantic classification and predict the 9 Degrees of Freedom (DoF) pose for both scan instances and CAD models, despite the differences in their low-level geometric features. Employing such a model enables us to utilize features acquired for classification in the 9-DoF object pose estimation task, while making the CAD retrieval and pose estimation process much faster.

2) Architecture Design: We experimented with two variations of the proposed architecture. In the first variant, we use 80 outputs for pose estimation, with 10 outputs allocated for each category. In this configuration, the pose output considered for pose loss calculation corresponds solely to the pose prediction corresponding to the predicted category, allowing for categorical separation of pose features in the network. In contrast, the second network employs only 10 output channels for pose, with the loss function being category-agnostic. Experimental comparisons consistently showed that the class-agnostic variant outperformed the category-aware one. This outcome is attributed to inaccuracies in classification predictions affecting pose predictions, as an incorrect category prediction may result in the selection of a pose vector for a category different from the object's actual category. Based on these outcomes, we chose the class-agnostic variant for our final results shown in Table I

To address object symmetries and the absence of symmetry annotations, we train this model utilizing a Chamfer distance-based loss function. The loss function's symmetry-agnostic nature enables it to offer more effective guidance for the pose estimation task. This approach streamlines the process by generating the 3D shape embedding and predicting the 9-DoF pose in a single forward pass, thereby significantly reducing the overall processing time.

D. VR Reconstruction

1) Human-in-the-loop: To enhance the authoring capabilities of CIS2VR, we have implemented a human-in-the-loop system before generating the final environment. This feature allows users to review and validate results at different stages in the pipeline. Specifically, prompts are presented to the user following instance segmentation, CAD model retrieval and alignment, and layout estimation steps. Users can choose to accept or reject the results generated by the framework at these steps, contributing to the reduction of inaccuracies and an overall improvement in the quality of the reconstructed scene. This system can also be disabled by the user to allow the framework to be executed without needing human intervention.

2) Semantic Reconstruction: Following the retrieval of CAD models corresponding to object instances in the scene and the replication of primary structural components, the subsequent phase involves the creation of a VR environment based on this acquired information. We chose Unity for our implementation due to the existing ecosystem for use cases like simulation, games, etc, but with very minor modifications, the framework can be made to work with other game engines as well. To summarize our design choices for different parts of the system (overview shown in Figure 1), we:

- Detect object instances in input RGB-D scans using a state-of-the-art 3D instance segmentation model.
- Generate shape encodings and predict the 9-DoF pose of each instance using our proposed 3D CNN model (class-agnostic variant). The shape encodings and predicted pose are used to retrieve and align a CAD model respectively.
- Estimate the layout of the scene using the predicted semantic mask and RANSAC.
- Use a human-in-the-loop approach to validate results from various steps of the pipeline.
- Construct a VR environment in Unity using retrieved CAD models, their corresponding pose estimations, and the structural components of the indoor scene.

IV. Implementation

A. Dataset

Owing to the challenges associated with manual annotations, real-world indoor RGB-D datasets remain limited, with notable examples including the Stanford 3D Indoor Scene Dataset [56], ScanNet v2 [36], and SUN RGB-D [47]. In addition to instance segmentation annotations, there is a requirement for annotations pertaining to 3D pose estimation to train our 9-DoF pose estimation algorithm. To the best of our knowledge, Scan2CAD [14] presented by Avetisyan et al. is the only dataset that satisfies our requirements. The dataset matches object instances from the ScanNet v2 dataset to CAD models from the ShapeNetV2 dataset, with annotations for pose and symmetry available for each matched CAD model.

1) 3D Pose Annotations: While the Scan2CAD dataset offers 3D pose annotations for instances in ScanNet v2, a direct mapping between object instances in ScanNet v2 and aligned CAD models in Scan2CAD is not explicitly provided. Additionally, not all object instances in ScanNet v2 possess corresponding pose annotations in Scan2CAD. Consequently, a need arises to establish a match between aligned CAD models from Scan2CAD and corresponding ScanNet object instances for the purpose of generating annotated data for 3D pose estimation. To address this challenge, we employ a matching procedure described in Algorithm 1 to match each pose annotation in Scan2CAD to the corresponding ScanNet

object instance. The instances matched to each aligned CAD model in Scan2CAD using Algorithm 1 are further validated by a human annotator. Subsequently, we update the Scan2CAD dataset with a matched ScanNet instance label for each aligned CAD model in the dataset. We provide this updated dataset as a part of the code that is made publicly available. In algorithm 1, the determination of the threshold for overlap is based on empirical studies. It is noteworthy that the pose annotations within this dataset are relative to the default poses of CAD models in the ShapeNet dataset. All CAD models in the ShapeNet dataset are centered at the origin and share a consistent default rotational alignment across each category, which we refer to as the default pose P_{def}.

Algorithm 1 Scan2CAD annotations to ScanNet instances

for S in ScanNet scenes **do**
  $anno \leftarrow$ Load scene annotation from Scan2CAD
  $insts \leftarrow$ Scene ScanNet instances
  $instCenters \leftarrow mean(i)$ for i in $insts$
  $cads \leftarrow$ Scan2CAD aligned CAD models
  $boxes \leftarrow$ CAD OBBs
  for i: $0 \rightarrow len(cads)$ **do**
    $c \leftarrow cads[i]$, $obb \leftarrow boxes[i]$
    $dists \leftarrow ||instCenters - obb.center||$
    $sortedInsts \leftarrow$ sort $insts$ based on closest $dists$
    for si in $sortedInsts$ **do**
      $overlap \leftarrow$ Overlap between obb and si
      $catMatch \leftarrow c.category == si.category$
      $d \leftarrow$ distance between c and si
      if $overlap > 80\%$ and $catMatch$ **then**
        $validMatch \leftarrow si$
        Remove si from $insts$
        break

2) Data Pre-processing: We use the pose annotations for scan objects to align them with P_{def}. During the training phase, a random pose, comprising of a random rotation, translation, and scale, is generated and applied to the object point clouds to introduce variance in training data. We formulate the pose of an object as a 10-element vector comprising of 3 coordinates for center, 3 coordinates for scale and rotation expressed in quaternion format (4 elements). We avoid using Euler angles for representing rotation as they are susceptible to ambiguity and Gimbal lock [61].

B. Instance Segmentation

Prominent recent contributions to 3D instance segmentation are evident in works such as [21] [22] [57] [49]. For our framework, we have opted to implement the approach proposed by Vu et al. [21] (SoftGroup). This decision is motivated based on their performance, ability to work on unprocessed point clouds, quality of code base, and ease of reproducing the published results. It's worth noting that, with ongoing advancements in 3D instance segmentation, this module can be readily substituted with a better performing algorithm in the future. SoftGroup builds upon the foundation laid by Chen et al. [49], introducing modifications and enhancements to the instance proposal pipeline. One notable limitation shared by many existing instance segmentation algorithms, including SoftGroup, pertains to the scope of supported semantic classes. SoftGroup, specifically, accommodates a total of 18 semantic classes, including structural categories such as walls, floor, ceiling, and windows, among others. Given our specific objective of replacing movable objects in the scene with CAD models to generate an interactive VR environment, we exclude object categories related to structural components from the CAD retrieval pipeline. Additionally, as we utilize a mixed dataset comprising ShapeNet and ScanNet objects, the semantic classes that can be employed are restricted to those common to both datasets. Consequently, this constrains the object categories supported by CIS2VR to 8: bathtub, bed, bookshelf, chair, desk, sofa, table, and toilet, presenting a significant limitation within the proposed framework.

Instance segmentation frameworks gauge their efficacy through variations of the intersection-over-union (IoU) metric, which assesses the overlap between ground truth and predicted masks. This metric, however, does not translate very well when extracting object instances from the predicted mask due to the presence of erroneously labeled small clusters of points in the predictions. Although these clusters may not significantly impact the mean IoU (mIoU) due to their size relative to the overall point cloud, they can substantially affect the quality of reconstruction. To mitigate the impact of these noisy predictions, we implement thresholds for the confidence score (Tconf) and the number of points in the instance (Tpoints). Our evaluation involves testing the algorithm with different configurations of T_{conf} and T_{points} to analyze the effects of varying these parameters on performance. The results of these evaluations are detailed in Figure 2. Based on our findings, we have identified T_{conf}=0.5 and T_{points}=512 as the optimal configuration for avoiding most false positives, which works best for our approach.

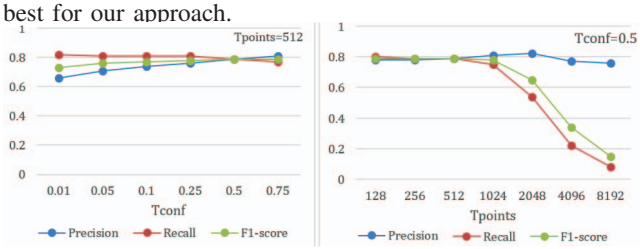

Fig. 2: Evaluating the effect of varying T_{conf} with $T_{points} = 512$ (left) and changing T_{points} with $T_{conf} = 0.5$ (right) on Precision, Recall and F1-score (Y axis).

C. Model retrieval and 9-DoF Pose estimation

Following instance segmentation, we retrieve and align a semantically and geometrically matching CAD model for each scanned object from our CAD database. To achieve this, we introduce a 3D CNN architecture inspired by the work of Choy et al. [19], which can encode the 3D shape of both scan and CAD objects, and predict the 9 Degrees of Freedom (DoF) pose of an object in a single forward pass.

1) Architecture: For our 3D shape encoding network, we train an object classification model and utilize the output of its final layers as our feature vector. Ideally, an object detection

model should be agnostic to the pose of the input object, meaning it should classify the object consistently regardless of its pose. Aligning with this objective, the features learned by such models are often pose-invariant, which contradicts our need to use the same network for predicting object pose. To overcome this challenge, we propose the network architecture illustrated in Figure 3, which is based on the Minkowski Engine framework proposed by Choy et al [19] and inspired by architecture proposed by Harry et al. [65]. The network extracts multi-scale point-wise features, which are then used in conjunction with more convolutional blocks to extract task-specific features. To facilitate the learning of both pose-invariant features for effective shape encoding and pose-dependent features for accurate pose estimation, we bifurcate the network into two branches. Introducing the fork earlier in the architecture enhances feature separation but increases model size, complexity, and inference time while delaying the fork compromises performance. After iterative testing, we settled on the final architecture, balancing lower model complexity without significant compromise in performance. The class prediction directly outputs from the linear layer of the classification branch, and the pose outputs pass through a TanH activation layer. The resulting model comprises approximately 45 million trainable parameters.

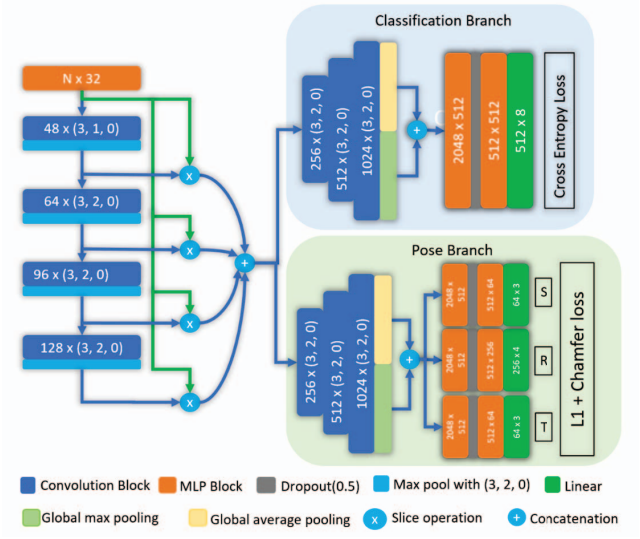

Fig. 3: Model architecture of the proposed retrieval and alignment method. The initial four convolutional blocks are responsible for extracting point-level geometric features across various scales, with later being branched off to pose estimation and shape classification modules.

2) Loss Function: To train the proposed network, we employ a hybrid loss function that accommodates both classification and pose estimation tasks. For the classification loss L_{cls}, our experimentation involved considering both Focal loss and Cross-Entropy (CE) loss, with CE loss demonstrating superior performance in most scenarios.

$$L_{cls} = BCE(X, X_{gt}) \tag{1}$$

For pose estimation, we use a combination of L1 loss (Ll1) and Chamfer loss (Lchamf). Although L2 loss has been widely used for regression tasks, we found the quadratic nature of the loss to be undesirable for loss values below 1, especially for rotation estimation. Our empirical studies also found L1 to be more effective for our architecture due to the linear nature of the loss.

$$L_{l1} = ||P_t - P_{gt\_t}|| + ||P_q - P_{gt\_q}|| + ||P_s - P_{gt\_s}|| \tag{2}$$

However, L1 loss lacks sensitivity to rotational symmetry. Although Scan2CAD contains symmetry annotations, only a minority of CAD models from ShapeNet are covered in the dataset. As we use all models from ShapeNet for our dataset, we choose to ignore the symmetry annotations for training. To account for rotational symmetry, we integrate a Chamfer distance-based loss function, a methodology relatively under-explored for objects, though previously applied in contexts such as human pose estimation [59] [60]. To compute Chamfer loss, we start by calculating a transformation matrix T using the predicted pose P and subsequently applying it to the default pose object point cloud, resulting in a transformed point cloud $\boldsymbol{S_1}$. We then calculate the Chamfer distance between $\boldsymbol{S_1}$ and the input point cloud $\boldsymbol{S_2}$ as:

$$L_{chamf}(\boldsymbol{S_1}, \boldsymbol{S_2}) = \frac{1}{|\boldsymbol{S_1}|} dist(\boldsymbol{S_1}, \boldsymbol{S_2}) + \frac{1}{|\boldsymbol{S_2}|} dist(\boldsymbol{S_2}, \boldsymbol{S_1}) \tag{3}$$

$$dist(\boldsymbol{S_1}, \boldsymbol{S_2}) = \sum_{P_1 \in \boldsymbol{S_1}} \min_{P_2 \in \boldsymbol{S_2}} ||P_1 - P_2||_2^2 \tag{4}$$

The final pose loss is computed as:

$$L_{pose} = L_{l1} + 5 * L_{chamf} \tag{5}$$

3) Training: The model is trained using a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01 and a momentum of 0.9. We employ Cosine Annealing (without warm restarts) to dynamically adjust the learning rate throughout the training process, and the network weights are randomly initialized. To investigate the hypothesis that low-level features learned for classification can enhance the learning of features for pose estimation, we conduct a two-step training procedure. Initially, the network undergoes pre-training on a classification task with L_{cls} for 10,000 iterations, involving both CAD and Scan objects. Subsequently, we freeze the initial convolution layers and the classification branch. The network is then trained for pose estimation using L_1 and L_{chamf} losses for 20,000 steps. This approach yields an 8% increase in average alignment accuracy compared to training the network for both pose estimation and classification tasks concurrently. Furthermore, it demonstrates a 10% increase in accuracy compared to training solely for pose estimation, with the classification branch completely removed. These findings support our hypothesis that lower-level features learned for classification contribute to the network's ability to acquire better features for pose estimation.

D. Layout detection

To recreate the VR scene's structural layout, we use the instance segmentation mask to extract points related to structural elements such as walls, floors, ceilings, counters, windows, and doors. RANSAC is employed for planar structure

TABLE I: Alignment results for CAD models on ScanNet data are assessed using pose annotations and evaluation metrics from Scan2CAD [14]. Numbers represent accuracy for each category, higher is better.

	bath	bookshelf	cabinet	chair	sofa	table	Class avg	Avg
FPFH [68]	0.00	1.92	0.00	10.00	5.41	2.04	3.22	4.45
SHOT [67]	0.00	1.43	1.16	7.08	3.57	1.47	2.45	3.14
Li et al. [66]	0.85	0.95	1.17	14.08	6.25	2.95	4.375	6.03
3D Match [35]	0.00	5.67	2.86	21.25	10.91	6.98	7.945	10.29
Scan2CAD [14]	36.2	36.4	34	44.26	70.63	30.66	42.025	31.68
End-to-End [64]	38.89	41.46	51.52	73.04	26.83	**76.92**	51.44	50.72
SceneCAD [15]	42.42	**36.84**	**58.33**	**81.23**	**82.86**	45.6	**57.88**	**61.24**
CIS2VR (Ours with GT instance annotations)	57.14	27.5	45.08	78.64	72.37	69.88	58.44	72.23
CIS2VR (Ours with SoftGroup)	**49.66**	19.52	29.92	67.47	54.02	56.54	46.19	60.25

detection, using parameters optimized for the ScanNet dataset (minimum points: 5, distance threshold: 10cm, iterations: 1000). The OBB around planar structures' points provides orientation information for replication in Unity. The absence of CAD models for structural components leads us to use a Unity unit cube model, deformed based on calculated pose parameters to replicate the layout. We also provide the user an option to replace the estimated layout (planes) with a mesh created from points corresponding to the layout from the input scan for a more accurate but less interactive environment.

Fig. 4: ScanNet RGB-D scenes (left) and corresponding Unity reconstruction (right), produced without any human intervention.

E. Unity Reconstruction

The associations between objects and CAD models, along with their respective transformation parameters for each input RGB-D scan, are written to a JSON file. This file is used by Unity to load CAD models and apply the corresponding transformation parameters. To enhance the visual similarity between the reconstructed scene and the RGB-D scan, we employ a coloring scheme for the aligned CAD models based on their corresponding scan objects. In this process, each vertex of the CAD model is colored according to the RGB values of points in the instance point cloud that are spatially closest to the vertex. Figure 4 shows VR scenes produced using input RGB-D scans from ScanNet.

V. Results and Evaluation

A. CAD Alignment

We evaluate our framework on the ScanNet Scan2CAD dataset, employing metrics introduced by Avetisyan et al. [14] to measure alignment accuracy. A successful alignment is defined within 20 cm translation, 20° rotation, and 20% scale of the ground truth. The evaluation encompasses the entire pipeline, including instance segmentation and object retrieval. To allow comprehensive comparison with existing works while quantifying the performance of the proposed pose estimation algorithm, we use two types of inputs for pose estimation: ground truth (GT) instance annotations and predicted instances from SoftGroup. Additionally, we restrict the evaluation to categories common to all works due to discrepancies in the supported object categories. Table I shows CIS2VR's performance benchmarked against methods utilizing hand-crafted feature descriptors (FPFH [68], SHOT [67], Li et al. [66]), learned feature descriptors (3D Match [35], Scan2CAD [14], End-to-End [64]), and recent techniques that leverage object-object and object-layout relationships for pose estimation (SceneCAD [15]). Note that these results are obtained with the human-in-the-loop system disabled.

The results indicate that our framework, including the instance segmentation module (SoftGroup), performs competitively, surpassing most existing approaches and closely trailing the current state-of-the-art approach by Avetisyan et al. SceneCAD [15]. Notably, when evaluating only the pose estimation algorithm using ground truth instance annotations from ScanNet, there is a significant improvement in performance. This suggests that the alignment performance of the framework is, to some extent, constrained by the effectiveness of the instance segmentation module. As advancements in indoor 3D segmentation continue, substituting this module with more advanced algorithms could further enhance the overall performance of the framework. The outcomes underscore the efficacy of our proposed model, coupled with a Chamfer loss-guided learning strategy, revealing that features learned by a model for a classification task contribute significantly to object pose estimation.

In our ablation studies, we investigated the influence of Chamfer distance on the algorithm's training. The removal of Chamfer distance from our loss formulation led to a noteworthy loss in performance, approximately 6%, underscoring the importance of the symmetry-agnostic nature of Chamfer loss in the learning process. Additionally, we explored the scenario of removing the classification branch and training the network exclusively for a class-agnostic pose estimation task. In this case, we observed a marginal decrease in performance, approximately 2%, compared to the architecture incorporating

the classification branch.

B. Runtime Analysis

To assess the runtime efficiency of our framework, we measured the time taken for different steps across the ScanNet validation dataset (consisting of 312 scenes). Table II displays the individual runtimes for various algorithms in the pipeline. The results indicate that the instance segmentation algorithm accounts for a significant portion of the overall runtime, while the combined time for shape encoding, pose estimation, and CAD model retrieval averages around 40ms. As shown in Figure 5, further examination of the instance segmentation algorithm's runtime across scenes with varying numbers of objects and spatial sizes reveals a general trend of increasing inference time based on the number of objects in the scene. In addition to the time taken for each step in Table II, we include the computational cost in terms of Trillion Floating Point Operations (TFLOP) for GPU-based tasks and CPU cycles for CPU-based tasks, offering a more objective quantification of the computational cost of the proposed pipeline. Our test system uses an RTX 3090 GPU (35.58 TFLOPS) with an AMD Ryzen 9 5900X CPU (3.7G cycles/s). Table III provides a comparison of the overall runtime of the proposed framework (including Unity scene generation time) in various scenarios with recent works on CAD model retrieval and alignment. Across the ScanNet test dataset consisting of 312 scenes, our framework takes approximately 0.68 seconds, on average, to convert an input RGB-D scan into a Unity VR scene. Notably, our framework demonstrates enhanced scalability with larger scenes containing more objects, efficiently converting a scene with 20 objects in just 0.66 seconds. It's worth noting that while the number of instances used by SceneCAD [15] (# objects 1, 5, 26) may not precisely match ours, [14] and [64], efforts were made to align them with the nearest corresponding number in our evaluation. The efficient runtime of the proposed framework enables its application in collaborative settings and real-time scenarios.

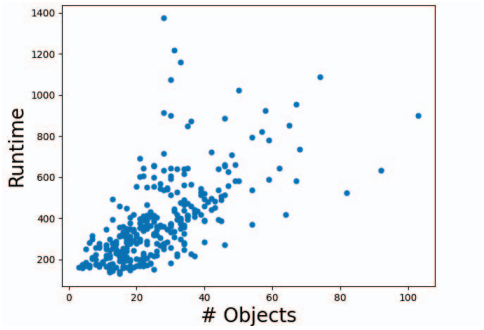

Fig. 5: Runtime on SoftGroup for validation scenes from ScanNet.

TABLE II: Runtime (in ms) & computational cost of modules in CIS2VR.

Task	Average Time taken	Computational cost
Instance segmentation	383.65	13.65 TFLOP
3D Shape Encoding + 9-DoF Pose estimation	28.88	1.027 TFLOPS
CAD model retrieval	10.84	40 Million Cycles
Layout estimation	60.91	225 Million Cycles
Unity scene generation	70.60	261 Million Cycles
Misc. processing	129.78	480 Million Cycles
Total	684.66	14.67 TFLOP + 10^9 CPU Cycles

TABLE III: Runtime (in ms) comparison with existing methods.

	7 Objects	16 Objects	20 Objects
Scan2CAD [14]	288.60	565.86	740.34
SceneCAD [15]	2.0(5)	-	2.60(26)
End-to-End [64]	0.62	1.11	2.60
CIS2VR (Ours)	0.55	0.61	0.66

VI. Conclusion

We have presented a VR authoring framework that allows users to create an interactive VR environment using an RGB-D scan in less than a second. To enable fast and accurate object retrieval and alignment, we propose a CNN-based architecture that jointly learns object classification and pose estimation, allowing it to leverage the features learned for the classification task for pose estimation, while significantly reducing the overall time taken for the whole process. We estimate the layout of the scene and approximate the texture of the CAD model (including structural components) based on corresponding scan objects to further enhance the visual similarity between the created environment and the RGB-D scan. The runtime efficiency of the proposed framework makes it a viable candidate for applications requiring collaboration and real-time interactions.

VII. Limitations and Future Work

While CIS2VR efficiently generates VR environments from RGB-D scans, our primary emphasis lies in object pose estimation. A notable limitation is the layout reconstruction algorithm, often resulting in disjointed structural components and incomplete scene layouts. Employing advanced techniques, such as room corner and edge detection, can enhance the quality and completeness of layouts. Improving CAD model retrieval quality involves leveraging datasets with CAD similarity annotations, like the Scan-CAD Object Similarity dataset [16]. The efficacy of our texture mapping depends on precise CAD model alignment; exploring advanced algorithms using key-point matching could enhance this aspect. Furthermore, investigating input modalities such as RGB-D videos and leveraging mature and accurate 2D object segmentation models holds the potential for increased flexibility.

Acknowledgment

This research was sponsored by the DEVCOM U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-21-2-0145 to B.P. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DEVCOM Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation.

REFERENCES

[1] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on User interface software and technology [Internet]. Santa Barbara California USA: ACM; 2011 [cited 2023 Aug 28]. p. 559–68. Available from: https://dl.acm.org/doi/10.1145/2047196.2047270

[2] Dai A, Nießner M, Zollhöfer M, Izadi S, Theobalt C. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. ACM Trans Graph. 2017 May 1;36(4):76a:1.

[3] Siddiqui Y, Thies J, Ma F, Shan Q, Nießner M, Dai A. RetrievalFuse: Neural 3D Scene Reconstruction with a Database [Internet]. arXiv; 2021 [cited 2023 May 15]. Available from: http://arxiv.org/abs/2104.00024

[4] Wang K, Zhang G, Bao H. Robust 3D Reconstruction With an RGB-D Camera. IEEE Trans Image Process. 2014 Nov;23(11):4893–906.

[5] Meerits S, Nozick V, Saito H. Real-time scene reconstruction and triangle mesh generation using multiple RGB-D cameras. J Real-Time Image Process. 2019 Dec 1;16(6):2247–59.

[6] Fu Y, Yan Q, Yang L, Liao J, Xiao C. Texture Mapping for 3D Reconstruction With RGB-D Sensor.

[7] Fu Y, Yan Q, Liao J, Xiao C. Joint Texture and Geometry Optimization for RGB-D Reconstruction. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Seattle, WA, USA: IEEE; 2020 [cited 2023 Oct 4]. p. 5949–58. Available from: https://ieeexplore.ieee.org/document/9156569/

[8] Kulkarni N, Johnson J, Fouhey DF. What's Behind the Couch? Directed Ray Distance Functions (DRDF) for 3D Scene Reconstruction [Internet]. arXiv; 2022 [cited 2023 Apr 13]. Available from: http://arxiv.org/abs/2112.04481

[9] Geiger, Andreas, et al. 'StereoScan: Dense 3d Reconstruction in Real-Time'. 2011 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2011,

[10] Nießner, Matthias, et al. 'Real-Time 3D Reconstruction at Scale Using Voxel Hashing'. ACM Transactions on Graphics, vol. 32, no. 6, Association for Computing Machinery (ACM), Nov. 2013, pp. 1–11,

[11] Whelan, Thomas, et al. 'ElasticFusion: Dense SLAM without A Pose Graph'. Robotics: Science and Systems XI, Robotics: Science and Systems Foundation, 2015,

[12] Han, Lei, et al. 'Real-Time Globally Consistent Dense 3D Reconstruction with Online Texturing'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, Institute of Electrical and Electronics Engineers (IEEE), Mar. 2022, pp. 1519–1533,

[13] Xu, Yabin, et al. 'HRBF-Fusion: Accurate 3D Reconstruction from RGB-D Data Using on-the-Fly Implicits'. ACM Transactions on Graphics, vol. 41, no. 3, Association for Computing Machinery (ACM), June 2022, pp. 1–19,

[14] Avetisyan, Armen, Manuel Dahnert, et al. 'Scan2CAD: Learning CAD Model Alignment in RGB-D Scans'. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019,

[15] Avetisyan, Armen, Tatiana Khanova, et al. 'SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans'. ArXiv [Cs.CV], 27 Mar. 2020, http://arxiv.org/abs/2003.12622. arXiv.

[16] Dahnert, Manuel, et al. 'Joint Embedding of 3D Scan and CAD Objects'. ArXiv [Cs.CV], 19 Aug. 2019, http://arxiv.org/abs/1908.06989. arXiv.

[17] Gupta, Saurabh, et al. 'Aligning 3D Models to RGB-D Images of Cluttered Scenes'. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015,

[18] Litany, Or, et al. 'ASIST: Automatic Semantically Invariant Scene Transformation'. Computer Vision and Image Understanding: CVIU, vol. 157, Elsevier BV, Apr. 2017, pp. 284–299,

[19] Choy, Christopher, Junyoung Gwak, et al. '4D SpatioTemporal ConvNets: Minkowski Convolutional Neural Networks'. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019,

[20] Pham, Quang-Hieu, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. "JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8827-8836. 2019.

[21] Thang Vu et al., "SoftGroup for 3D Instance Segmentation on Point Clouds," arXiv:2203.01509 [Cs], March 2, 2022, http://arxiv.org/abs/2203.01509.

[22] Jiang, Li, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. "PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation." In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4866–75. Seattle, WA, USA: IEEE, 2020.

[23] Ngo, Tuan Duc, Binh-Son Hua, and Khoi Nguyen. "ISBNet: A 3D Point Cloud Instance Segmentation Network with Instance-Aware Sampling and Box-Aware Dynamic Convolution." In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13550–59. Vancouver, BC, Canada: IEEE, 2023.

[24] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 303–312. ACM, 1996.

[25] Newcombe, Richard A, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. "KinectFusion: Real-Time Dense Surface Mapping and Tracking," n.d.

[26] Sun, Jiaming, et al. 'NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video'. ArXiv [Cs.CV], 1 Apr. 2021, http://arxiv.org/abs/2104.00681. arXiv.

[27] Qi, Charles R., et al. 'PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation'. ArXiv [Cs.CV], 2 Dec. 2016, http://arxiv.org/abs/1612.00593. arXiv.

[28] Han, Lei, et al. 'OccuSeg: Occupancy-Aware 3D Instance Segmentation'. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020,

[29] Song, Shuran, and Jianxiong Xiao. 'Sliding Shapes for 3D Object Detection in Depth Images'. Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 634–651,

[30] Li, Yangyan, et al. 'Database-Assisted Object Retrieval for Real-Time 3D Reconstruction'. Computer Graphics Forum: Journal of the European Association for Computer Graphics, vol. 34, no. 2, Wiley, May 2015, pp. 435–446,

[31] Qi, Charles R., Li Yi, et al. 'PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space'. ArXiv [Cs.CV], 7 June 2017, http://arxiv.org/abs/1706.02413. arXiv.

[32] Qi, Charles R., Hao Su, et al. 'Volumetric and Multi-View CNNs for Object Classification on 3D Data'. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016,

[33] Park, Jeong Joon, et al. 'DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation'. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019,

[34] Mescheder, Lars, et al. 'Occupancy Networks: Learning 3D Reconstruction in Function Space'. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019,

[35] Zeng, Andy, et al. '3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions'. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017,

[36] Dai, Angela, et al. 'ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes'. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017,

[37] Chang, Angel X., et al. 'ShapeNet: An Information-Rich 3D Model Repository'. ArXiv [Cs.GR], 9 Dec. 2015, http://arxiv.org/abs/1512.03012. arXiv.

[38] Shapira, Lior, and Daniel Freedman. 'Reality Skins: Creating Immersive and Tactile Virtual Environments'. 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2016,

[39] Manni, Alessandro, et al. 'Snap2cad: 3D Indoor Environment Reconstruction for AR/VR Applications Using a Smartphone Device'. Computers & Graphics, vol. 100, Elsevier BV, Nov. 2021, pp. 116–124,

[40] Ipsita, Ananya, et al. 'VRFromX: From Scanned Reality to Interactive Virtual Experience with Human-in-the-Loop'. Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, ACM, 2021,

[41] Kari, Mohamed, et al. 'TransforMR: Pose-Aware Object Substitution for Composing Alternate Mixed Realities'. 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, 2021,

[42] Lim, J. J., Pirsiavash, H., & Torralba, A. (2013). Parsing ikea objects: Fine pose estimation. In Proceedings of the IEEE international conference on computer vision (pp. 2992-2999).

[43] Izadinia, H., Shan, Q., & Seitz, S. M. (2017). Im2cad. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5134-5143).

[44] Huang, S., Qi, S., Zhu, Y., Xiao, Y., Xu, Y., & Zhu, S. C. (2018). Holistic 3d scene parsing and reconstruction from a single rgb image. In Proceedings of the European conference on computer vision (ECCV) (pp. 187-203).
[45] Choi, Sungjoon, et al. 'Robust Reconstruction of Indoor Scenes'. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015,
[46] Armeni, Iro, et al. '3D Semantic Parsing of Large-Scale Indoor Spaces'. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016,
[47] Shuran, Samuel P., and Jianxiong Lichtenberg. 'Sun Rgb-d: A Rgb-d Scene Understanding Benchmark Suite'. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 567–576.
[48] Rukhovich, Danila, et al. 'FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection'. Lecture Notes in Computer Science, Springer Nature Switzerland, 2022, pp. 477–493,
[49] Chen, Shaoyu, et al. 'Hierarchical Aggregation for 3D Instance Segmentation'. ArXiv [Cs.CV], 4 Aug. 2021, http://arxiv.org/abs/2108.02350. arXiv.
[50] Nan, Liangliang, et al. 'A Search-Classify Approach for Cluttered Indoor Scene Understanding'. ACM Transactions on Graphics (TOG), vol. 31, no. 6, 2012, pp. 1–10.
[51] Wu, Zhirong, et al. '3D ShapeNets: A Deep Representation for Volumetric Shapes'. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015,
[52] Song, Shuran, et al. 'Semantic Scene Completion from a Single Depth Image'. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017,
[53] Deitke, Matt, et al. 'RoboTHOR: An Open Simulation-to-Real Embodied AI Platform'. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020,
[54] Zheng, Jia, et al. 'Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling'. Computer Vision ECCV 2020, Springer International Publishing, 2020, pp. 519–535,
[55] Roberts, Mike, et al. 'Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding'. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021,
[56] Armeni, Iro, et al. '3D Semantic Parsing of Large-Scale Indoor Spaces'. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016,
[57] Liang, Zhihao, et al. 'Instance Segmentation in 3D Scenes Using Semantic Superpoint Tree Networks'. ArXiv [Cs.CV], 17 Aug. 2021, http://arxiv.org/abs/2108.07478. ArXiv.
[58] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation" (arXiv, June 15, 2016), http://arxiv.org/abs/1606.04797.
[59] Wei Liu, Wei Fan, and Yuan He, "Collaborated Directional Chamfer Matching for 3D Hand Pose Estimation," n.d.
[60] Mugalodi Rakesh et al., "Aligning Silhouette Topology for Self-Adaptive 3D Human Pose Recovery" (arXiv, 2022),
[61] Evan G. Hemingway and Oliver M. O'Reilly, "Perspectives on Euler Angle Singularities, Gimbal Lock, and the Orthogonality of Applied Forces and Applied Moments," Multibody System Dynamics 44, no. 1 (September 1, 2018): 31–56,
[62] Laurent Zwald and Sophie Lambert-Lacroix, "The BerHu Penalty and the Grouped Effect" (arXiv, July 30, 2012),
[63] Julian Straub et al., "The Manhattan Frame Model—Manhattan World Inference in the Space of Surface Normals," IEEE Transactions on Pattern Analysis and Machine Intelligence 40, no. 1 (January 2018): 235–49,
[64] Armen Avetisyan, Angela Dai, and Matthias Niessner, "End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, 2019), 2551–60,
[65] Harry Pratt et al., "FCNN: Fourier Convolutional Neural Networks," in Machine Learning and Knowledge Discovery in Databases, ed. Michelangelo Ceci et al., vol. 10534, Lecture Notes in Computer Science (Cham: Springer International Publishing, 2017), 786–98,
[66] Li, Y., Dai, A., Guibas, L., Nießner, M.: Database-assisted object retrieval for realtime 3D reconstruction. In: Computer Graphics Forum, vol. 34, pp. 435–446. Wiley Online Library (2015)
[67] Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010).
[68] Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation ICRA'2009, pp. 3212–3217. Citeseer (2009)