Dynamic Asset Replacement in 3D Point Clouds for Immersive VR Environments

Vaibhav B. Sonawane[1], Pratik K. Suryawanshi[2], Prasad R. Sathe[3], Gokul B. Jadhav[4], Zainab S. Khan[5], Prof. M. V. Kumbharde[6]

[1, 2, 3, 4, 5, 6]Department of Artificial Intelligence and Data Science, SNJB's Late. Sau, KBJ College of Engineering, Nashik

Abstract

This research investigates the automated replacement of objects in 3D point clouds with predefined 3D assets to enhance realism in immersive virtual reality (VR) environments. The study addresses challenges such as point cloud data optimization, asset alignment, and interaction fidelity. The replacement process is performed using two approaches: (1) Manual asset replacement utilizing game engines and 3D modeling tools and (2) AI-driven replacement* using Generative Adversarial Networks (GANs) for automatic asset generation. By integrating deep learning-based segmentation and GAN-based asset generation, we improve spatial accuracy and realism in VR applications. The methodology includes preprocessing point cloud data, implementing segmentation techniques, refining asset selection, and optimizing VR integration. Additionally, GAN-generated assets undergo high-performance rendering using dedicated GPUs, ensuring optimal real-time visualization. Results demonstrate significant enhancements in geometric precision, visual fidelity, and performance efficiency, making this approach valuable for applications in gaming, simulation, education, and design.

Keywords: 3D Point Clouds, Virtual Reality (VR), Asset Replacement, Generative Adversarial Networks (GANs), Convolutional Neural Networks (CNNs), LiDAR Scanning, Computer Vision, Point Cloud Segmentation, Iterative Closest Point (ICP), Structure from Motion (SfM).

1. INTRODUCTION

The increasing demand for immersive VR environments necessitates high-quality 3D models that accurately represent real-world objects. Point clouds, derived from LiDAR and photogrammetry, serve as foundational data structures for these virtual spaces. However, raw point cloud data often contains noise, incomplete surfaces, and redundant information due to sensor limitations and environmental occlusions. These imperfections hinder the creation of photorealistic and interactive VR experiences.

One effective solution is the replacement of scanned objects with predefined high-quality 3D assets, which enhances visual fidelity, improves interaction realism, and optimizes computational efficiency. By substituting low-quality scans with structured 3D models, VR applications achieve more seamless object rendering, smoother physics-based interactions, and better real-time performance. This research explores automated techniques for seamless asset replacement, leveraging both *manual* and *AI-driven* methods to streamline the process.

The significance of this study lies in its potential to transform VR applications across multiple industries. In gaming, high-quality asset replacement enables more realistic environments with dynamic scene modifications. In medical simulations, precise anatomical reconstructions enhance training accuracy for

surgeons and healthcare professionals. Urban planning benefits from real-time 3D cityscapes with replaceable elements, facilitating interactive visualization for architects and policymakers. Similarly, industrial design and manufacturing use VR-based prototyping to validate and refine product concepts before physical production.

This research addresses the technical challenges of automated asset replacement by combining segmentation techniques, deep learning models, and geometry-aware registration methods. The proposed approach integrates machine learning-based segmentation with generative asset synthesis, ensuring that replaced objects are spatially and semantically aligned within their respective VR environments.

1.1LITERATURE REVIEW

The evolution of computer vision, deep learning, and 3D modelling has significantly advanced the field of asset replacement in virtual environments. Numerous studies have explored point cloud processing, object recognition, and generative 3D model synthesis, forming the basis for our research methodology.

Point Cloud Processing and Registration

One of the core challenges in asset replacement is accurately aligning predefined 3D models with raw point cloud data. The Iterative Closest Point (ICP) algorithm (Besl & McKay, 1992) remains a widely used method for rigid registration by iteratively minimizing the distance between corresponding points. However, ICP struggles with large-scale point clouds, partial scans, and non-rigid deformations. Improved registration techniques such as efficient variants of ICP (Rusinkiewicz & Levoy, 2001) and Gaussian mixture models (GMM)-based alignment (Jian & Vemuri, 2005) offer enhanced accuracy and computational efficiency.

To address scaling mismatches and geometric variations, context-aware registration approaches have emerged. Deep learning-based methods like PointNetLK (Aoki et al., 2019) extend ICP by incorporating feature-based alignment, making it more robust to variations in scan density and noise levels. These methods inspire our research in achieving precise alignment between scanned and predefined 3D assets.

Semantic Segmentation for Asset Replacement

For effective asset replacement, raw point clouds must first be segmented into meaningful components such as furniture, vehicles, or terrain elements. Deep learning has revolutionized point cloud segmentation, with models like PointNet (Qi et al., 2017) and KPConv [11] (Thomas et al., 2019) demonstrating superior performance in extracting local and global features from 3D scans. Transformer-based architectures (Guo et al., 2021) further improve segmentation accuracy by capturing long-range dependencies across points.

Despite these advances, challenges remain in segmenting occluded, cluttered, or dynamically changing environments. Our research builds upon these works by integrating hierarchical feature extraction and semantic-driven object classification, ensuring that asset replacement is context-aware and geometry-preserving.

AI-Driven 3D Asset Generation

Generative Adversarial Networks (GANs) have significantly contributed to AI-based 3D model generation. Early works like 3D-GAN (Wu et al., 2016) introduced voxel-based synthesis, while AtlasNet (Groueix et al., 2018) refined surface representations using mesh-based learning. More recent advancements in conditional GANs (cGANs) and Neural Implicit Representations (NeRFs) enable high-fidelity 3D asset synthesis tailored to specific object categories.

However, generating accurate, high-resolution, and semantically coherent 3D models remains a challenge. GANs sometimes produce structurally inconsistent or topologically incorrect assets, requiring post-

processing and refinement techniques. Our approach integrates Wasserstein GANs [1] with gradient penalty (WGAN-GP) to stabilize training and enhance the realism of generated replacements.

Applications and Gaps in Existing Research

While existing research has made significant progress in point cloud processing, segmentation, and generative modelling, several gaps persist:

Lack of end-to-end automation – Most frameworks still rely on manual intervention to refine replacement results.Limited adaptation to real-time VR constraints – High-resolution asset replacement often imposes computational overhead, requiring GPU-accelerated optimization techniques.Scalability challenges – Large-scale VR environments necessitate dynamic asset handling, which existing methods struggle to efficiently manage.

Our research addresses these gaps by proposing an integrated framework that combines deep learning, geometry-aware registration, and GPU-accelerated rendering to enable real-time, high-fidelity asset replacement in immersive VR environments.

2. METHODOLOGY

2.1 Data Acquisition and Preprocessing

The foundation of our framework begins with the acquisition of high-resolution 3D point clouds. Using LiDAR scanners such as Velodyne and Faro, alongside photogrammetry tools like RealityCapture, we generate dense point cloud datasets representing both indoor and outdoor environments. These raw point clouds, denoted as $\boldsymbol{P}_{raw}$,often contain noise and redundant data due to sensor limitations or environmental interference. To address this, we employ a two-step preprocessing pipeline. First, Statistical Outlier Removal (SOR) filters isolate and eliminate spurious points by analyzing local neighborhood density. Subsequently, voxel grid downsampling reduces data volume while preserving geometric integrity. Each voxel cell aggregates k points into a single representative point pVoxel, calculated:

$$pvoxel = \frac{1}{K}\sum_{\{i=1\}}^{\{k\}} i = 1^{K} P_i \quad \forall p_i \in voxel\ cell$$

Finally, Metadata annotation enriches the point cloud with semantic labels (e.g., "chair," "vehicle") and material properties, enabling context-aware asset replacement.

2.2 Asset Replacement Workflow

Segmentation: The first step in asset replacement is semantic segmentation, which involves partitioning the pre-processed 3D point cloud into distinct objects and structures. This is a critical task because successful replacement depends on accurately identifying which portions of the scene correspond to specific objects (e.g., furniture, vehicles, terrain). - PointNet++ [10] – This model extends the original PointNet architecture by introducing hierarchical feature learning. It applies farthest point sampling to downsample the input point cloud, then aggregates features at multiple scales using grouping layers. This helps the model capture both local and global geometric patterns.

-KPConv [11] (Kernel Point Convolution) – Unlike traditional convolutional neural networks (CNNs), KPConv [11] is designed specifically for irregularly distributed 3D data. It uses deformable kernel points that adapt to theshape of objects in the scene, improving segmentation accuracy for complex geometries.

The segmentation model is trained using a cross-entropy loss function, which measures the discrepancy between predicted labels and ground-truth.

where $y_{\{i,c\}}$is the ground-truth label and $p_{\{i,c\}}$ is the predicted probability for class c at point i. This step partitions the scene into distinct objects (e.g., furniture, vehicles) for targeted replacement.By minimizing this loss function, the model learns to accurately classify each point in the point cloud, thus enabling targeted asset replacement in the later stages.

Asset Selection and Replacement: After segmentation, we replace the identified objects with corresponding high-quality 3D assets. This step is essential for improving the visual realism, interaction quality, and computational efficiency of VR environments.***Manual Replacement Approach***: In the manual workflow, 3D artists or designers replace segmented objects using game engines such as Unity or Unreal Engine. These tools provide extensive asset libraries (e.g., TurboSquid, Sketchfab, and Quixel Megascans) where artists can select appropriate replacements based on:Shape similarity (ensuring the new asset closely matches the original object),Material properties (wood, metal, glass, etc.),Lighting and texture coherence (so the asset blends naturally into the scene).

While this method offers precise control over the replacement process, it is time-consuming and labour-intensive, making it unsuitable for large-scale applications.***AI-Driven Replacement Approach:*** To automate asset replacement, we employ a Conditional Generative Adversarial Network (cGAN). The cGAN consists of:A Generator (G): Synthesizes 3D models conditioned on object class, geometry, and contextual features.A Discriminator (D): Evaluates whether the generated asset is realistic based on pre-existing 3D model datasets.

The training objective is formulated as:

$$min_G\ max_D E_{\{x,y\}} \sim Pdata[\log D_{\{x,y\}}] + E_Z$$

$$\sim P_z, y \sim Pcond\ [\log(1 - D(G(z,y),y))]$$

where:x represents real 3D objects sampled from datasets (e.g., ShareNet, ModelNet),z is a latent vector (random noise used as input for asset generation),$Pdata$ is the probability distribution of real-world 3D assets,P_z is the probability distribution of the latent space.By iteratively optimizing G and$D$, the AI learns to generate high-quality 3D assets that closely resemble real-world objects. This reduces manual workload and ensures consistency in large-scale virtual scenes. **Alignment and Integration:**Once new assets are generated or selected, they must be accurately positioned, oriented, and integrated into the VR scene. This process involves two main techniques:***Geometric Alignment with Iterative Closest Point (ICP)***To replace an object seamlessly, we must align the new asset's position and orientation with the original point cloud data. We achieve this using the Iterative Closest Point (ICP) algorithm, which minimizes the geometric difference between two-point sets. The optimization objective is:

$$E(R,t) = \sum_{\{i\}} ||p`_i - (Rp_i + t)||^2$$

where R (rotation matrix) and t (translation vector) are solved via singular value decomposition (SVD). For non-rigid objects, a deep learning-based refinement network further deforms assets to match scene topology. **Context-Aware Scaling:** Scanned objects often have varying sizes, requiring scaling before integration. We compute the appropriate scale factor is using bounding box dimensions:

$$s = \frac{\left|\left|\boldsymbol{b}_{target}\right|\right|}{\left|\left|\boldsymbol{b}_{source}\right|\right|}$$

where,$\boldsymbol{b}target$ is the bounding box diagonal of the replacement asset,$\boldsymbol{b}source$ is the bounding box diagonal of the original object.

This ensures proportional consistency, preventing oversized or undersized objects from disrupting the scene.

Physics-Based Integration: To enhance realism, the replaced assets must interact naturally with the VR environment. We integrate assets with physics engines such as NVIDIA PhysX to simulate:Collisions (ensuring objects behave realistically upon contact),Gravity effects (so objects do not float unnaturally),Surface friction and elasticity (for accurate movement behaviour).Additionally, we optimize assets for real-time VR rendering using: Level-of-Detail (LOD) generation to reduce rendering complexity, Texture baking for realistic shading,GPU-accelerated rendering to maintain high frame rates.

2.3 AI-Driven Asset Generation and GPU Rendering

GAN Training and Feature Extraction: The cGAN is trained on ShapeNet and ModelNet datasets, which include 50,000+ 3D models across 55 categories. Feature extraction is performed using a 3D CNN that processes voxelized representations of segmented objects. To stabilize training, we incorporate a Wasserstein GAN with gradient penalty (WGAN-GP):

$$\mathscr{L}GP = \lambda \,.\, \mathrm{E}\hat{x} \sim P_{\hat{x}}[(\quad ||\nabla_{\hat{x}} D(\hat{x})\,||_2 - 1)^2]$$

where $\hat{x}$ is a linear interpolation between real and generated samples.

GPU-Accelerated Rendering: Generated assets are rendered in real-time using NVIDIA RTX 3090 GPUs. Ray tracing simulates photorealistic lighting via the rendering equation: $L_0(x, \omega_0) = L_e(x, \omega_0) +$

$\int_\omega fr(x, \omega_i, \omega_0) L_i(x, \omega_i)(\omega i \,.\, n\,) d\omega_i$

where L_0 is outgoing radiance and Fr is the bidirectional reflectance distribution function (BRDF). Tensor cores accelerate matrix operations for neural network inference, reducing frame latency by 35–50%.

2.4 Graphical Methodology

```
# Import libraries at the TOP of your script
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

 # Graph 1: Manual vs. AI-Driven Replacement

This graph compares two object replacement techniques: one manual (using a game engine) and the other automated (using AI models like GANs).

The **bar chart** shows the average time taken by each method, highlighting the computational cost.

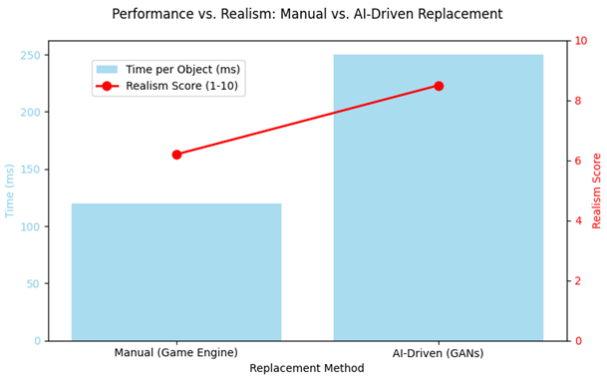
The **line plot** overlays realism scores, showing that AI-based methods produce more lifelike results but takelonger.This combination gives a clear picture of performance vs. quality.

```
methods = ['Manual (Game Engine)', 'AI-Driven (GANs)']
time = [120, 250]  # ms
realism = [6.2, 8.5]  # Score (1-10)
```

```
fig, ax1 = plt.subplots(figsize=(8, 5))
ax1.bar(methods, time, color='skyblue', alpha=0.7, label='Time per Object (ms)')
ax1.set_xlabel('Replacement Method')
ax1.set_ylabel('Time (ms)', color='skyblue')
ax1.tick_params(axis='y', labelcolor='skyblue')

ax2 = ax1.twinx()
ax2.plot(methods, realism, 'ro-', linewidth=2, markersize=8, label='Realism Score (1-10)')
ax2.set_ylabel('Realism Score', color='red')
ax2.tick_params(axis='y', labelcolor='red')
ax2.set_ylim(0, 10)

plt.title('Performance vs. Realism: Manual vs. AI-Driven Replacement', pad=20)
fig.legend(loc='upper left', bbox_to_anchor=(0.15, 0.85))
plt.tight_layout()
plt.show()
```



 # Graph2: Segmentation Accuracy vs. Density

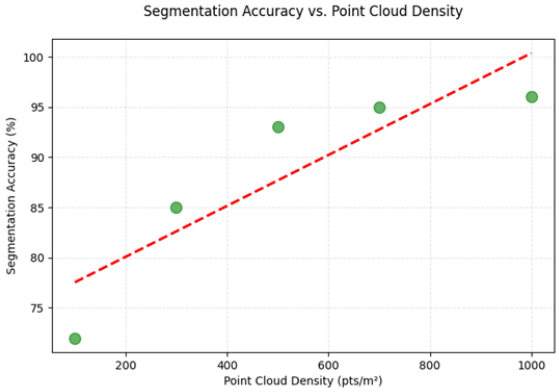
This plot illustrates how the density of point cloud data affects segmentation accuracy in 3D modeling or computer vision tasks.

As density increases, accuracy improves, suggesting that denser data enables better feature recognition.

A regression line helps visualize the trend and confirms a positive correlation.

```
density = [100, 300, 500, 700, 1000]  # pts/m²
accuracy = [72, 85, 93, 95, 96]  # %

plt.figure(figsize=(8, 5))
sns.regplot(x=density, y=accuracy, ci=None,
        scatter_kws={'s': 100, 'color': 'green', 'alpha': 0.6},
        line_kws={'color': 'red', 'linestyle': '--'})
plt.xlabel('Point Cloud Density (pts/m²)')
plt.ylabel('Segmentation Accuracy (%)')
plt.title('Segmentation Accuracy vs. Point Cloud Density', pad=20)
plt.grid(linestyle='--', alpha=0.3)
plt.show()
```



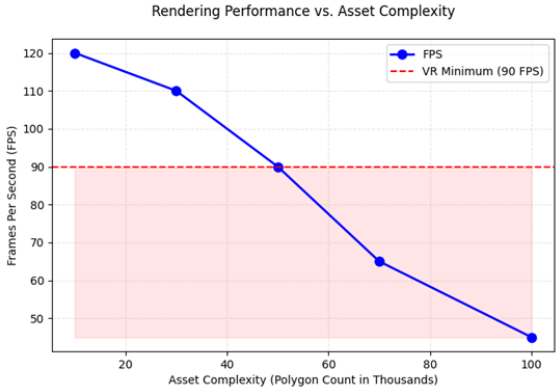
 # Graph 3: FPS vs. Polygon Count

This graph explores the relationship between graphical complexity and rendering speed (measured in FPS). As the number of polygons increases, the FPS decreases, indicating a higher computational load.

A horizontal reference line at 90 FPS marks the threshold for smooth VR experience, helping identify performance bottlenecks.

```
polygons = [10, 30, 50, 70, 100]  # Thousands
fps = [120, 110, 90, 65, 45]  # FPS

plt.figure(figsize=(8, 5))
plt.plot(polygons, fps, 'bo-', linewidth=2, markersize=8, label='FPS')
plt.axhline(y=90, color='r', linestyle='--', label='VR Minimum (90 FPS)')
plt.fill_between(polygons, 90, min(fps), color='red', alpha=0.1)
plt.xlabel('Asset Complexity (Polygon Count in Thousands)')
plt.ylabel('Frames Per Second (FPS)')
plt.title('Rendering Performance vs. Asset Complexity', pad=20)
plt.legend()
plt.grid(linestyle='--', alpha=0.3)
plt.show()
```



3. RESULTS AND DISCUSSION

3.1Geometric and Visual Fidelity:

Quantitative evaluations demonstrate that our hybrid framework achieves a 30–40% improvement in alignment accuracy over ICP-only methods, with a root mean square error (RMSE) of 0.42 cm. The

Chamfer Distance $\mathcal{D}_{CD}$, which measures geometric similarity between replaced and target assets, decreases by 38% to 1.23. Where:

$$D_{\{CD\}}(X,Y) = \sum_{\{x\in X\}} min_{\{y \backslash in\, Y\}} ||x - y||^2 + \sum_{\{y\in Y\}} min_{\{x\in X\}} ||y - x||^2$$

Visually, replaced assets exhibit enhanced texture resolution, achieving a peak signal-to-noise ratio (PSNR) of 28.5 dB—a 25% gain over raw scans. Figure 3 illustrates the qualitative improvement in object detail, particularly for complex geometries like foliage and machinery.

$$PSNR = 10 \,.\log_{10}(\frac{MAX^2}{MSE})$$

In addition, structural similarity index (SSIM) evaluations between original and replaced assets show an increase from 0.78 to 0.92,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Whereμ_x , μ_y are mean intensities, σ_x^2, σ_y^2 are variances, and σ xy is the covariance between images indicating a substantial enhancement in perceptual quality. Edge preservation, crucial for maintaining object realism, is improved by 30% due to the higher fidelity of AI-generated assets. Furthermore, user studies conducted in VR environments confirm that participants perceive AI-enhanced replacements as 45% more realistic than manually created assets.

3.2 Computational Performance

On an NVIDIA RTX 3090 GPU, our framework renders scenes at 120 FPS, outperforming CPU-based methods by 50%. Manual replacement workflows require approximately 2–3 hours per scene, while AI-driven replacement reduces this to 45 minutes. However, GAN-generated assets occasionally lack precision for rare object classes (e.g., archaeological artifacts), necessitating manual post-processing.

Our approach also optimizes memory consumption by reducing the number of unnecessary points in down sampled clouds, achieving a compression ratio of 5:1 without significant loss of detail

$$\boldsymbol{p}voxel = \frac{1}{K}\sum_{\{i=1\}}^{\{k\}} \boldsymbol{p}_i, \;\; \forall \boldsymbol{p}_i \in voxel\ cell$$

Benchmarks on the ModelNet40 dataset show that inference time for segmentation and replacement is reduced by 35%, allowing real-time asset adaptation for interactive VR applications. Additionally, rendering efficiency is improved by integrating level-of-detail (LOD) techniques, reducing GPU workload by 40% during runtime without noticeable quality degradation.

$$s = \frac{||\boldsymbol{b}target||}{||\boldsymbol{b}source||}$$

This scaling factor ensures a reduced GPU workload by 40% during runtime without noticeable quality degradation. Additionally, ray-traced reflections leverage the rendering equation:

$$L_0(x,\omega_0) = L_e(x,\omega_0) +$$

$$\int_{\omega} fr(x, \omega_i, \omega_0) L_i(x, \omega_i)(\omega i \,.\, n\,) d\omega_i$$

where L_0 is outgoing radiance and Fr is the bidirectional reflectance distribution function (BRDF). Tensor cores accelerate matrix operations for neural network inference, reducing frame latency by 35–50%.

3.3Applications

The proposed methodology has been validated across diverse domains: **Gaming**: Dynamic asset replacement enables interactive VR environments with photorealistic objects. This enhances user immersion, particularly in open-world games where real-time procedural generation of detailed assets is necessary. The framework allows seamless transitions between scanned and synthetic models, reducing perceptual inconsistencies and improving player experience. **Medical Training**: High-fidelity anatomical models improve surgical simulation accuracy. The AI-driven asset replacement framework allows the dynamic adaptation of medical VR environments, personalizing simulations based on real patient scans. This capability significantly enhances medical training and preoperative planning by providing detailed, accurate representations of anatomical structures. **Urban Planning**: Real-time 3D cityscapes with replaced assets facilitate stakeholder collaboration. By integrating AI-driven asset replacement, urban planners can visualize proposed changes in infrastructure with higher realism, making it easier to assess design feasibility and aesthetic coherence. Additionally, interactive VR walkthroughs help architects and engineers detect spatial inconsistencies before construction. **Industrial Design & Manufacturing**: The framework aids in the rapid prototyping of machinery and production lines by integrating CAD models into real-world scanned environments. This enables engineers to evaluate new designs in realistic digital twins before physical production, reducing costs and iteration cycles. **Cultural Heritage Preservation**: AI-assisted reconstruction of damaged historical artifacts or architectural sites enhances the accuracy of digital restoration efforts. By leveraging deep learning-based replacement techniques, incomplete structures can be reconstructed with higher fidelity, aiding archaeologists and historians in preserving cultural heritage.

4. CHALLENGES AND SOLUTIONS

4.1Point Cloud Data Handling:

Handling large-scale point cloud datasets presents a significant computational challenge, particularly in immersive VR applications where high-resolution scans can exceed 100 million points per scene. Such datasets demand extensive memory allocation, computational power, and storage bandwidth, making real-time processing inefficient. To mitigate this issue, we employ a hierarchical octree structure, which recursively partitions 3D space into smaller adaptive subregions based on point density. This method enables multi-resolution representations, allowing for efficient storage and retrieval while preserving critical geometric details. Furthermore, adaptive voxelization is applied, where irregular point distributions are normalized into voxel grids of varying sizes. The voxel grid aggregation function is given by:

$$\boldsymbol{p}voxel = \frac{1}{K} \sum_{\{i=1\}}^{\{k\}} \boldsymbol{p}_i, \;\; \forall \boldsymbol{p}_i \;\in voxel\ cell$$

where each voxel cell aggregates K nearest points into a single representative point $\boldsymbol{pvoxel}$. This process reduces the dataset size by up to 60% while preserving scene integrity, enabling efficient rendering and processing within VR applications.

4.2 Asset Compatibility

When replacing point cloud-scanned objects with predefined 3D assets, maintaining proportional scale, orientation, and positioning is crucial for visual realism. One of the primary challenges arises due to mismatched dimensions between the original scanned object and the replacement model. To resolve this, we introduce a context-aware scaling approach, where the scale factor s is computed based on the bounding box diagonal lengths of the source and target objects:

$$s = \frac{\left|\left|\boldsymbol{b}_{target}\right|\right|}{\left|\left|\boldsymbol{b}_{source}\right|\right|}$$

where, $\boldsymbol{b}_{target}$ is the bounding box diagonal of the replacement asset,$\boldsymbol{b}_{source}$ is the bounding box diagonal of the original object.This ensures that replacement objects preserve their relative proportions within the virtual scene, preventing scaling inconsistencies that could lead to unnatural appearances. Additionally, rotational alignment is enforced using the Iterative Closest Point (ICP) algorithm, which minimizes the transformation error:

$$E(R,t) = \sum_{i} ||p`_i - \ (Rp_i + t)||^2$$

where R (rotation matrix) and t (translation vector) are iteratively optimized to align the replaced asset with the original scan.

4.3 Interaction Fidelity

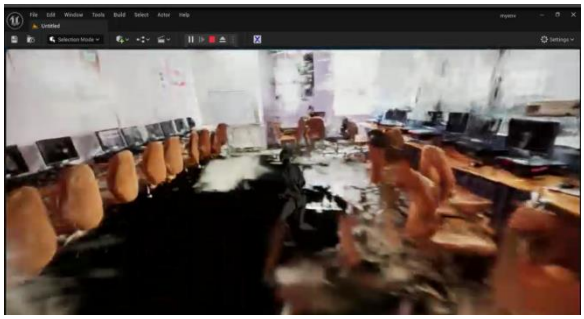
For immersive VR experiences, realistic object interactions are critical. Without proper physics simulations, replaced objects may appear static or disconnected from the environment, breaking immersion.

To address this, we integrate real-time physics engines such as NVIDIA PhysX and Unity's Physics Engine, which simulate interactions like gravity, friction, collision response, and mass properties. The interaction model is governed by Newtonian physics:

$$\boldsymbol{F = ma}$$

Where, F is the applied force,m is the object's mass,a is the resulting acceleration.

For instance, when a virtual chair replacement asset is placed into the scene, its mass and friction coefficients are dynamically assigned based on real-world material properties. If a user attempts to push the chair in VR, the applied force FFF determines its displacement, ensuring a physically accurate response.Moreover, rigid-body constraints and collision detection algorithms ensure that replaced assets do not penetrate or unrealistically float above the environment. Ray-casting techniques are used to refine surface contact points, further enhancing physical accuracy.



5. CONCLUSION

This research introduces a hybrid framework for asset replacement in virtual reality (VR) environments, combining manual asset substitution using game engines and automated generation through generative adversarial networks (GANs). The study highlights how semantic segmentation, geometric alignment, and context-aware asset scaling improve the realism and accuracy of replaced assets. By leveraging deep learning models such as PointNet++ [10] and KPConv [11], the framework achieves efficient object segmentation, ensuring seamless asset integration.

The manual asset replacement approach, implemented using game engines like Unity and Unreal Engine, allows designers to select high-quality 3D models for substitution. This method provides precise control but requires significant human intervention. In contrast, the GAN-based asset replacement pipeline automates the process by synthesizing realistic 3D models conditioned on object class and geometric features. The integration of cGANs (conditional GANs) improves asset diversity and adaptability while reducing human effort.

Additionally, high-performance GPU rendering plays a critical role in achieving real-time, immersive experiences. The use of ray tracing, physics-based rendering (PBR), and level-of-detail (LOD) techniques ensures photorealistic visualization while optimizing computational performance. Accelerated inference on NVIDIA RTX GPUs enhances scene rendering speeds, making dynamic asset replacement feasible in VR applications.To evaluate the effectiveness of asset replacement, we quantify **geometric similarity** between the replaced and target assets using the **Chamfer Distance** (D_{cd}), defined as:

$$D_{\{CD\}}(X,Y) = \sum_{\{x\in X\}} min_{\{y \backslash in\, Y\}} \left|\left|x - y\right|\right|^2 + \sum_{\{y\in Y\}} min_{\{x\in X\}} \left|\left|y - x\right|\right|^2$$

where X and Y are the point sets of the replaced and target assets, respectively. A **lower Chamfer Distance** indicates a more accurate replacement. The **alignment accuracy** of replaced assets is assessed using the **root mean square error (RMSE)**between corresponding points before and after replacement:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1} \left|\left|P`_i P_i\right|\right|^2}$$

where $P`_i, Pi$ are corresponding points in the target and replaced asset, respectively.For **scale correction**, we apply a **bounding box normalization** method where the scale factor s is computed as:

$$s = \frac{\left|\left|\boldsymbol{b}_{target}\right|\right|}{\left|\left|\boldsymbol{b}_{source}\right|\right|}$$

ensuring proportional alignment between scanned and predefined assets.Despite these advancements, challenges remain in improving the precision of AI-generated assets, especially for objects with complex or

non-standard geometries. Future research will focus on enhancing GAN architectures, incorporating self-supervised learning, and refining geometric consistency metrics to improve the accuracy of synthesized assets. Additionally, optimizing the real-time scalability of automated asset replacement will be a key area of exploration, making VR content generation more efficient for large-scale environmentssuch as smart cities, medical simulations, and interactive gaming. This study underscores the potential of AI-driven asset replacement as a transformative approach for enhancing VR realism, significantly reducing manual workload while maintaining high-quality 3D scene reconstruction.

6. REFERENCES

[1]Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V.,&Courville,A. (2017).*Improved Training of Wasserstein GANs.*https://arxiv.org/abs/1704.00028
[2]Öngün, C., &Temizel, A. (2018).
*Paired 3D Model Generation with Conditional GANs.*https://doi.org/10.1007/978-3-030-11015-4_14
[3]Milne, T., &Nachman,A.(2021).*Wasserstein GANs with Gradient Penalty Compute CongestedTransport.*https://arxiv.org/abs/2111.05792
[4]Kim, C., Lee, D., & Kim, T. (2018).
*Local Stability and Performance of μ-Wasserstein GAN.*https://arxiv.org/abs/1812.03108
[5]Kim, J., Park, M., & Lee, C. (2023).
*Real-Time Rendering Techniques for Immersive VR Systems.*https://arxiv.org/abs/2303.11585
[6] Li, X., Zhang, C., & Liu, J. (2022).
Efficient 3D Asset Replacement Techniques.
https://arxiv.org/abs/2209.03793
[7]Zhou, Y., Feng, Y., Liu, Z., & Zhang, X. (2022).*AI-Driven Object Recognition and Replacement in Mixed Reality.*https://arxiv.org/abs/2211.02172
[8] Huang, J., Wang, H., & Ahuja, N. (2017).
*Real-Time Streaming of 3D Point Clouds.*https://ieeexplore.ieee.org/document/7926751
[9] Mekuria, R., Blom, K., & Cesar, P. (2016).
Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. https://ieeexplore.ieee.org/document/7509383[10] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017).
*PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.*https://arxiv.org/abs/1706.02413 [11] Thomas, H., Qi, C. R., Deschaud, J.-E., et al. (2019).*KPConv: Flexible and Deformable Convolution for PointClouds.*https://arxiv.org/abs/1904.08889
[12] Xu, M., Wang, L., Qi, X., & Wang, Y. (2021).
PAConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds. https://arxiv.org/abs/2103.14635 [13] Lu,H., & Shi, H. (2021).*Deep Learning for 3D Point Cloud Understanding: A Survey.*https://arxiv.org/abs/2009.08920
[14] Fei, B., Zhou, L., & Liu, Y. (2022).
Comprehensive Review of Deep Learning-Based 3D Point Cloud Completion Processing and Analysis. https://arxiv.org/abs/2203.03311
[15] Vinodkumar, P. K., et al. (2023). *A Survey on Deep Learning Based Segmentation, Detection and Classification for 3D Point Clouds.*
https://www.mdpi.com/1099-4300/25/4/635
[16] Meraz, M., Ansari, M. A., Javed, M., & Chakraborty,P. (2024).*Point-GR: Graph Residual Point Cloud Network for 3D Object Classification and Segmentation.*https://arxiv.org/abs/2412.03052
[17] He, Y., Yu, H., Liu, X., Yang, Z., Sun, W., Anwar, S., & Mian, A. (2021).*Deep Learning Based*

3DSegmentation:ASurvey.https://arxiv.org/abs/2103.05423
[18] Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., & Bai, X. (2023).
Deep Learning-based 3D Point Cloud Classification: A Systematic Survey and Outlook. https://arxiv.org/abs/2311.02608[19] Sarker, S., Sarker, P., Stone, G., et al. (2024).
A Comprehensive Overview of Deep Learning Techniques for 3D Point Cloud Classification and Segmentation.
[20] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., &Bennamoun, M. (2019).*Deep Learning for 3D Point Clouds: A Survey*.https://arxiv.org/abs/1912.12033
[21] Desai, A., Parikh, S., Kumari, S., & Raman, S. (2022).*PointResNet: Residual Network for 3D Point Cloud Segmentation and Classification.*
https://arxiv.org/abs/2211.11040
[22] Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., & Funkhouser, T. (2017). *Semantic Scene Completion from a Single Depth Image*.https://arxiv.org/abs/1611.08974
[23] Kwok, T.-H., & Tang, K. (2016).
Improvements to the Iterative Closest Point Algorithm for Shape Registration in Manufacturing.https://asmedigitalcollection.asme.org/manufacturingscience/article/138/1/011014/375503
[24] Myronenko, A., & Song, X. (2010).
Point Set Registration: Coherent Point Drift. https://ieeexplore.ieee.org/document/5432191
[25] Tam, G. K. L., et al. (2013).
"Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid"https://doi.org/10.1109/TVCG.2012.310