IntelliCap: Intelligent Guidance for Consistent View Sampling

Ayaka Yasunaga [1*] Hideo Saito [1] Dieter Schmalstieg [2] Shohei Mori [2,1†]

[1] Keio University [2] University of Stuttgart

Figure 1: Intelligent guidance for consistent view sampling using a combination of AR and AI tools. (a) Our system visualizes areas that lack view samples for spatial and angular coverage to synthesize view-dependent effects (the pink-white stripes and sphere, respectively). (b) The user is encouraged to erase the pink-white stripes by pointing the smartphone camera at the visualizations. (c) Meanwhile, our system identifies objects that potentially need denser samples evaluated by LLM, and generates spherical proxies to encourage the operator to take more images around them. (d) Our approach can avoid exhaustive exploration, yet guarantee the final rendering quality.

Abstract

Novel view synthesis from images, for example, with 3D Gaussian splatting, has made great progress. Rendering fidelity and speed are now ready even for demanding virtual reality applications. However, the problem of assisting humans in collecting the input images for these rendering algorithms has received much less attention. High-quality view synthesis requires uniform and dense view sampling. Unfortunately, these requirements are not easily addressed by human camera operators, who are in a hurry, impatient, or lack understanding of the scene structure and the photographic process. Existing approaches to guide humans during image acquisition concentrate on single objects or neglect view-dependent material characteristics. We propose a novel situated visualization technique for scanning at multiple scales. During the scanning of a scene, our method identifies important objects that need extended image coverage to properly represent view-dependent appearance. To this end, we leverage semantic segmentation and category identification, ranked by a vision-language model. Spherical proxies are generated around highly ranked objects to guide the user during scanning. Our results show superior performance in real scenes compared to conventional view sampling strategies.

Index Terms: Visual guidance, 3D Gaussian splatting, semantic information, large language model, view sampling.

1 Introduction

New methods for novel view synthesis from images have made great progress. For example, 3D Gaussian Splatting (3DGS) and neural radiance fields (NeRF) synthesize photorealistic images at high frame rates [12, 20]. The underlying scene representations are optimized using differentiable rendering. A key requirement for differentiation is uniform and dense sampling of the input views, so that a reliable estimation of gradients can be performed.

Unfortunately, human camera operators often lack the skill or motivation to acquire such uniform and dense samples. Time limitations may prevent enough images from being collected. Individuals with limited knowledge of view synthesis may not understand the required strategy for image taking. After a lengthy scene optimization process, which can take tens of minutes to hours, it may not be possible to continue image acquisition. The severity grows with the size of the scene that should be acquired. While existing strategies work well for scanning individual objects [3, 19], solutions that scale to larger scenes or open areas are lacking [18].

Instant visual guidance using various forms of augmented reality (AR) has been proposed to assist individuals in image acquisition. The guidance tool can visually indicate where to take more photos or show intermediate results of the ongoing scanning and reconstruction progress. For example, 3D annotations are popular for ensuring aliasing-free light field rendering in the near field based on plenoptic sampling theory [18], but do not trivially scale to larger areas. Volumetric reconstruction [22] and 3DGS SLAM [16] show which parts of the scene are already covered. However, they are only able to cover large areas by trading off resolution and view-dependent properties for robustness and performance.

We aim to provide a unified solution for guiding image acquisition that works on multiple scales and supports complex scenes without sacrificing quality (Figure 1). We automatically identify objects and determine the required amount of image coverage matching the object's topology and material properties. We rely on AI tools to segment objects and estimate their visual complexity, such as topology, reflective, and diffractive materials. A vision model, Detectron2 [34], performs object segmentation and seman-

*e-mail: ayaka.yasunaga@keio.jp
†e-mail: s.mori.jp@ieee.org

Table 1: Qualitative comparisons of visual guidances for novel view synthesis.

Method	Operator	Supported scene scale	View-dep. sampling and rendering	Visual guidance
ActiveSplat [15]	Robot	Room (2D locations with 360 images)	No (SplaTAM [11])	None
ULF [3]	Human	Single object	Yes (ULF [3])	3D sphere
MRLF [19]	Human	Single object	Yes (ULF [3])	3D sphere
LLFF [18]	Human	Forward facing scenes	Yes (MPI [18])	3D indicators on 2D grid
FS2MPI [9]	Human	Forward facing scenes	Yes (MPI [18])	2D rectangle
Ours	Human	3D open-space with multiple objects	Yes (3DGS [12] and NeRF [31])	Progressive 3D scene mesh and spheres

tic categorization of objects. Semantic categories are ranked using a large language model (LLM) to inform the image-taking process.

With this information, spherical indicators are shown in AR to support the user's sampling procedure. We adapt the AR visualization to the object scales and scene geometry to avoid suggesting unreachable areas. To enable operation on mobile devices, our system can offload online vision recognition tasks from mobile devices to a server by transmitting selected keyframes from the view samples.

We validate our approach by comparing it with a conventional guidance approach that determines occupancy and information gain of a 3D reconstruction. To evaluate our work, we develop an evaluation scheme for this new task to compare multiple 3DGS datasets collected by different participants in open environments.

The contributions of this work can be summarized as follows:

- We propose an AR guidance system for quality view synthesis that uses semantic categorization of objects seen in input views and LLM processing to determine scanning requirements of observed objects.
- We present two situated visualizations in combination, consisting of a scene (e.g., temporal 3D reconstruction) and objects abstracted as spheres, for comprehensive spatial and angular view sampling.
- We demonstrate superior performance in 3DGS and Nerfacto view synthesis using view samples from our approach through a user study and rendering quality assessment. To achieve this, we developed an evaluation scheme to assess this new task of view sampling with AR guidance for view synthesis.

2 Background and Related Work

We discuss the limitations of current visual guidance approaches for view sampling in open 3D scenes and elaborate on our key ideas to compare them with existing solutions in related areas.

2.1 Background

The challenge of visual guidance for view sampling lies in indicating where to sample views without having access to the final view synthesis results. Table 1 summarizes the current approaches.

Small area coverage Current solutions either rely on geometric scanning [15], which lacks view-dependent sampling, or are based on the well-established plenoptic sampling theory [2]. The latter, which ensures anti-aliased view synthesis, applies mainly to forward-facing cameras due to its assumption of a fixed scene depth range [18]. However, in real-world open scenes, the minimum and maximum depths vary depending on the scene and the view frustum. Thus, the applicability of the theory is limited.

Earlier 3D scene representations, such as multi-plane images (MPI) [18, 9], require hundreds of images to cover even a square meter of space. Their visual guidance systems focus on precisely aligning the camera with 3D indicators at predefined positions. In contrast, modern approaches such as 3D Gaussian splatting (3DGS) can cover significantly larger areas with the same number of images, enabling new possibilities for visual guidance in open environments. Despite this improvement, the current best practice remains to capture as many images as possible to conquer scene complexity.

Missing knowledge of scene objects While view-dependent effects must be thoroughly captured, data acquisition should be completed in a practical timeframe. Therefore, the sampled areas must be prioritized. For example, glass bottles or leafy plants demand denser sampling than simple planar objects like desks. Novice users often lack intuition for how many images to take until sampling is sufficient. They lack a visual guidance system that can signal the completion of the sampling and suggest attention to areas of interest. However, scoring individual objects during photo capture remains a challenge.

Our solution To tackle these two technical challenges, we propose (1) progressive visual indicators using 3D mesh and sphere proxies, and (2) LLM-based scoring of detected objects and their categories. Users walk freely through the environment, while our system performs 3D reconstruction to evaluate spatial coverage. Meanwhile, the system detects objects and generates spherical proxies to attract user attention (Figure 1b). As users continue scanning, new objects may be found, spawning additional proxies to prompt denser view sampling. Once all highlighted visualizations disappear, the user can consider the view-sampling process complete. Alternatively, users may keep expanding the sampled areas by searching for the remaining attractions in the environment. This is naturally supported by the progressive operation of our system.

2.2 Visual guidance for view sampling

AR can visually indicate where to operate a camera and collect image samples [18, 9]. Popular forms of annotations include 3D axes [18], 2D planes [9, 1], and hemispheres [3, 19] surrounding a target area. These methods recast the photographing task as a data collection task. For example, when using a smartphone, the 3D annotations can be directly overlaid on the viewfinder display. The user must then navigate the phone to "intersect" [18, 9, 1] it with the 3D annotations or "cast a ray" [3, 29, 19] hitting the annotation. When the geometric constraints indicated to the user are met, the smartphone automatically captures the image and advances to the next annotation. The user only has to collect all images indicated by the annotation, which often changes color to indicate which parts have been fulfilled already.

During capture on the mobile device, the annotations are derived from theoretical bounds such as the required sampling rate. Performing even a preliminary analysis-by-synthesis to steer the feedback would be too costly in these circumstances. Again, typical approaches place 3D annotations at the locations according to the plenoptic sampling theory [2, 24]. The locations are fixed at the beginning of the photo session. This process involves a detailed scan of the scene before starting the guidance, so sampling rates can be guaranteed [18, 9, 1]. This process is rather rigid and is not suitable for spontaneous capture sessions. Existing approaches aim at capturing only one target object at a time [3, 1] or a small area in front of the operator [18, 9, 15].

Our approach dynamically evolves the capture area by detecting unobserved regions and newly discovered objects that require

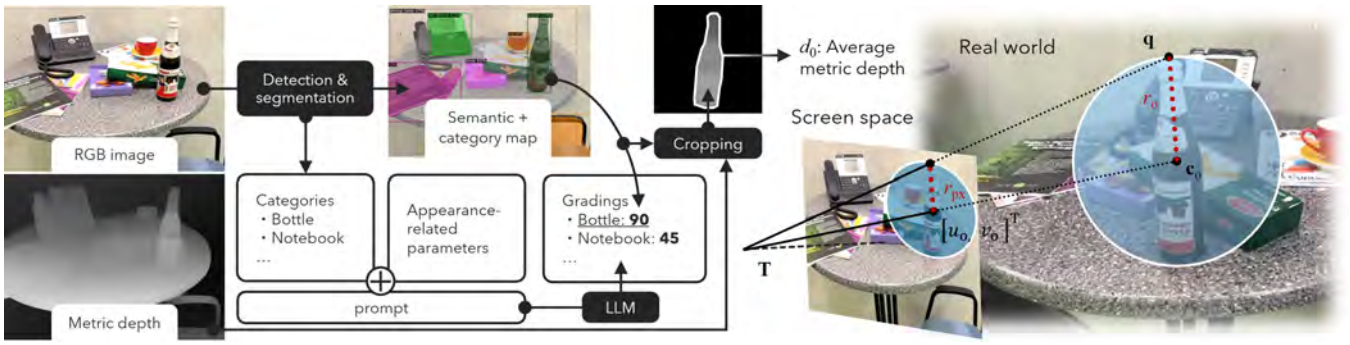

Figure 2: 3D sphere generation. Given a pair of RGB and metric depth maps, our system detects important objects for view sampling and projects them into real-world space as spheres. We use an LLM with a custom prompt template to quantify the visual characteristics of the detected objects and prioritize them accordingly. Objects with high scores are anchored in the 3D scene to indicate areas where the user should focus their sampling efforts.

denser sampling. It combines standard 3D scanning techniques with spherical proxies to provide intuitive visual cues suited for open scenes, while leveraging computer vision to automate scene understanding. This enables adaptive and progressive visualization, relieving users from the burden of manually assessing scene complexity and coverage.

2.3 Real-time view synthesis as visual feedback

Real-time reconstruction can provide immediate visual feedback on rendering quality, allowing users to continue sampling new views until satisfied. The state-of-the-art online reconstruction of neural view synthesis hardly meets real-time requirements due to its costly optimization process [39, 38], except when a dedicated view synthesis pipeline is used [5]. The latest 3DGS reconstruction methods run at interactive frame rates, but prune details for increased speed and robustness [17, 11]. For full-quality reconstruction, batch optimization must be performed later on [11].

A substitute for the full reconstruction used in newer commercial applications relies on real-time volumetric reconstruction (cf. KinectFusion [22] and DTAM [23]). However, these volumetric reconstructions use a coarse resolution and are mostly used to estimate scene coverage and occlusion, but they cannot ensure that sufficient geometric or photometric details are captured.

To compensate for the missing fine detail during capture, we introduce spherical proxies that indicate areas that require additional angular view samples. We implement this approach on a smartphone and demonstrate its effectiveness compared to standard methods that highlight missing regions in real-time reconstruction.

2.4 Next best view

Research on uncertainty analysis in neural rendering and 3DGS has targeted floater removal [6], compression [7] and decision making about the next best view [10]. To find the next best view samples online, geometric information gain is evaluated [15] by rendering 360 views at candidate locations and selecting the most empty view. Next-best-view finding involves on-demand scene optimization, which is affected by the same computational limits as applied to SLAM systems. Reported results rely on simulated environments assuming a pre-scanned mesh [4], hours of intermediate training [30], or systematic dense uniform scanning [13], all of which is not applicable for real-time scanning sessions operated by humans.

Our system is designed for fully online operation, offloading computer vision processes to a server and utilizing pre-computed object importance priors provided by an LLM. The smartphone is primarily responsible for 3D scanning, visualization, and photo capture. This lightweight configuration enabled our user study. While our system does not provide explicit navigation to the next-best-view locations, it allows users to identify them through intuitive visualizations.

3 Method

We developed a system that captures images at fixed time intervals as the user walks through the environment. To guide the user, the system provides two types of visual highlights in the 3D scene: one based on spatial coverage, and another based on angular coverage for objects that may impact view synthesis quality due to their photometric complexity (Figure 1a). These visualizations are progressively updated and refined as view sampling progresses (Figure 1b). The collected view samples are then used to optimize 3DGS for rendering the corresponding scenes (Figure 1c).

3.1 Spatial and angle coverage analysis

3D mesh proxy for spatial coverage Existing methods estimate scene coverage [15, 28, 39] by constructing a 3D point cloud. Surfaces with sparse or missing points are identified as under-sampled and targeted for densification. In our approach, we highlight regions that have not yet been reconstructed, based on the current state of the 3D reconstruction (Figure 1a). Incomplete areas are overlaid with pink-white stripes, while the rest is taken from the live camera feed. This provides visual feedback about which areas have already been visited or not. For simplicity and performance in implementation, we utilize the smartphone's native scene meshing API (ARCore or ARKit).

3D sphere proxy for angular coverage While the mesh proxy supports spatial coverage, it does not account for angular sampling. Users may miss critical photometric details, such as specular reflections or transparency, that vary significantly with viewpoint and are known to be challenging to reconstruct without sufficient angular diversity [13]. To address this, we detect objects likely to exhibit such photometric complexity during the walkthrough. Rather than identifying all objects that can easily occupy the scene with spheres, we selectively detect only those that benefit most from denser view sampling. Only for these, we generate 3D sphere proxies [3, 19].

Each sphere proxy consists of a 3D position $\mathbf{c}$, a radius r, a set of subsurface $S \in \{s_i | 0 \leq i \leq N\}$, as $(\mathbf{c}, r, S)$. Each subsurface corresponds to a specific view sample, allowing us to control the sampling density and coverage speed. As users capture new views, the corresponding subsurfaces are marked as covered and disabled by setting their transparency to full. The subsurfaces are retained throughout the session. The marked subsurface becomes transparent to tell the capture to complete as in Figure 1a.

Figure 3: Scene adaptive 3D sphere visualization. (a) Scene-adaptive rendering guides the user and prevents them from walking into unreachable areas. (b) Spheres are only displayed within a valid depth range illustrated in orange to avoid sampling from positions too close or too far. (c) Two nearby spheres are merged into a single sphere, inheriting the properties of the older one. This naturally supports the integration of multi-view inputs for the same object from different viewpoints.

In the following sections, we describe how these 3D spheres are generated and how their representations are dynamically updated.

3.2 3D sphere generation

Figure 2 provides an overview of the 3D sphere generation process. The input to our guidance system consists of multiple captured views, each comprising an RGB image, a metric depth map, and intrinsic and extrinsic camera parameters. These input views are processed independently. We use Detectron2 [34] to detect object segments and their categories from RGB images. The depth maps are used to estimate object size in the 3D scene. Identified objects are then analyzed to determine whether they should be registered for denser view sampling. Since it is typically unclear from a single view how geometrically or photometrically complex an object is, we use object semantics to bridge this gap.

The semantics in the form of object categories are processed with an LLM to determine an estimate of the visual object complexity. The key idea is to use a foundation model, which contains common human knowledge, to obtain properties that are difficult to quantify. Therefore, we preferred a closed vocabulary to provide the necessary knowledge efficiently. We create a prompt template and recursively ask to score the scores for objects' appearance characteristics. The template prompt is defined as follows, '`I have a dataset containing object IDs and their corresponding categories.`

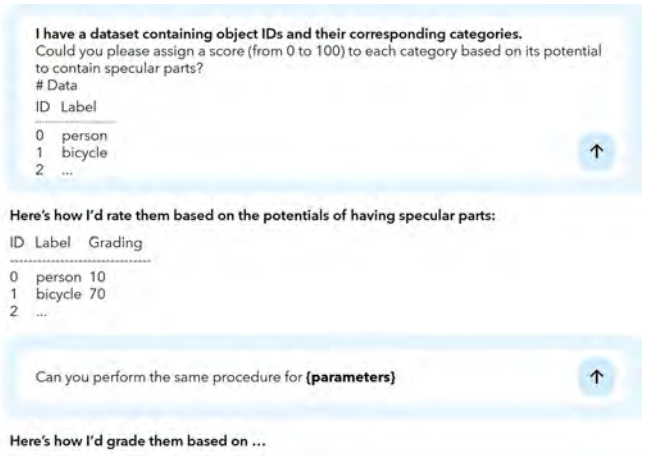

Figure 4: The prompt for scoring the object categories to quantify their appearance characteristics.

`Could you please assign a score (from 0 to 100) to each category based on its potential to contain {parameter}? {object list},'` where `{parameter}` $\in$ `{geometric complexity, texture complexity, size, specularity, transparency}`. The questions for the rest of the parameters in a sequence to keep the context (Figure 4). The category labels are pre-scored by an LLM (e.g., MS Copilot[1]). The scores of all metrics are averaged for the final score. Once an object is found that exceeds a threshold, the object is tagged for higher angular sampling.

The tagged objects are approximated with 3D spheres. The sphere visualizes the coverage, indicating from which viewing angles the object was already observed [3, 19]. In our approach, subsurfaces are evenly distributed on the sphere for uniform capture [25]. The spheres are anchored to the real objects. We assume that the center of the sphere is at the centroid of the segmented object in screen space $[u_o, v_o]^\intercal$ with associated depth coordinate d_o, which is computed as the average depth of the object samples (after outlier removal). We transform the sphere center $\mathbf{c}_o = [x_o, y_o, z_o]^\intercal = \mathbf{R}\mathbf{K}^{-1} d_o [u_o, v_o, 1]^\intercal + \mathbf{t}$ into world coordinates using the camera position $\mathbf{t}$, rotation matrix $\mathbf{R}$, and intrinsic matrix $\mathbf{K}$. The radius of the sphere in pixels $r_{px} = k \max(u_{max} - u_{min}, v_{max} - v_{min})$ is calculated from the 2D bounding box of the object, which has the corners $[u_{min}, u_{max}]^\intercal$ and $[v_{min}, v_{max}]^\intercal$. A scaling factor k ensures that the sphere fully encloses the object. To convert the radius into metric units, we assume an auxiliary point $\mathbf{q} = \mathbf{R}\mathbf{K}^{-1} d_o [u_o, v_o - r_{px}, 1]^\intercal + \mathbf{t}$ located above the sphere center. The metric radius of the new sphere is determined with the help of $\mathbf{q}$ as $r_o = ||\mathbf{q} - \mathbf{c}||$.

3.3 Dynamically adjusting 3D spheres

Occlusion handling To prevent confusion about occluded areas, we avoid overlaps between spheres and scene objects, helping users understand that they do not need to capture views behind occluders such as walls. The alpha value (transparency) of each sphere is dynamically adjusted based on its depth relationship with surrounding scene elements. Specifically, if the depth of the sphere is closer than the depth of the scene at a given pixel, the sphere is displayed. Otherwise, it is hidden behind the scene geometry.

To mitigate potential misalignment caused by the imperfect scene mesh under reconstruction, we use real-time depth streams from the smartphone. In addition, we render soft edges at the intersections between spheres and scene geometry to provide visual tolerance. The level of sphere transparency is determined by the offset in depth $\Delta d/t$, where Δd is the depth difference between the sphere surface and the scene, and t is a tolerance threshold set to 5.0 cm by default (Figure 3a).

[1] https://copilot.microsoft.com/

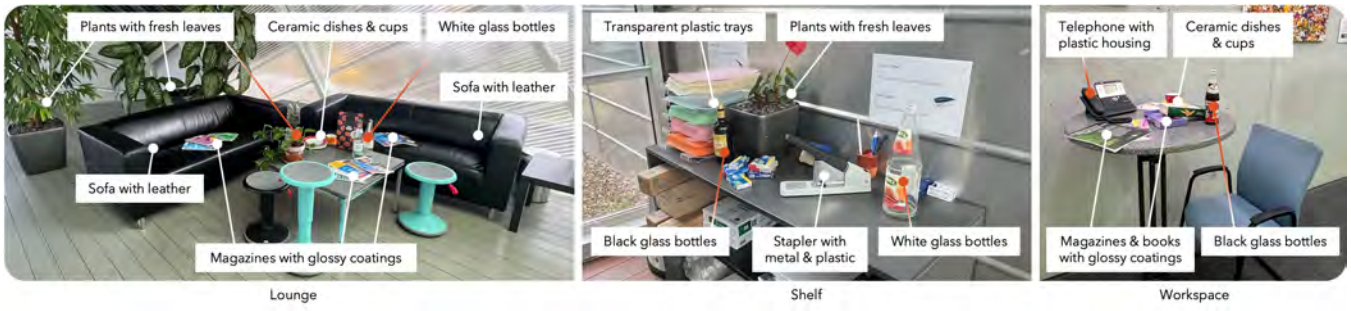

Figure 5: Large (≈ 9 m$^2 = 3 \times 3$), medium (≈ 2 m$^2 = 2 \times 1$), and small (≈ 1 m$^2 = 1 \times 1$) scenes for the experiments. To increase visual complexity, we intentionally arranged white and black glass bottles, plastic bottles, ceramic cups and dishes, plants with fresh leaves, books and magazines with glossy coatings, and a telephone with plastic housing. The objects highlighted with red lines have high potential to be abstracted as spherical proxies if they are appropriately detected as one of the complex objects of labels: `vase`, `bottle`, `cellphone`, etc.

Distance-based suppression To keep the object of interest within the user's field of view (FoV), spheres are visualized only when they fall within a valid distance range (Figure 3b). When outside this range, a fixed-size dot is shown instead, indicating the location of an object identified by the system. View sampling cannot be completed unless the user approaches the sphere within the required distance, naturally encouraging closer interaction. We allow the full sphere to appear once it occupies more than 20% of the camera FoV. When it reaches 100%, the sphere disappears, indicating that the user is too close and that view sampling at this range is unnecessary.

Merging spheres As the number of spheres increases, the workload for capturing becomes higher. Therefore, our objective is to minimize the number of spheres by intelligently merging expendable spheres (Figure 3c). Initially, a collision check is performed between the spheres, and only intersecting spheres are considered for merging. In the trivial case where a sphere is completely inside another sphere, the smaller sphere can simply be discarded. Otherwise, the two spheres $(\mathbf{c}_1, r_1, S_1)$ and $(\mathbf{c}_2, r_2, S_2)$ are merged into a new sphere $(\mathbf{c_{new}}, r_{\text{new}}, S_{\text{new}})$. We compute the center of the new sphere as the midpoint between the two sphere centers, $\mathbf{c}_{\text{new}} = (\mathbf{c}_1 + \mathbf{c}_2)/2$ and the radius as the sum of the radii plus the distance between the sphere centers, $r_{\text{new}} = (r_1 + r_2 + ||\mathbf{c}_1 - \mathbf{c}_2||)/2$. The radius of the new sphere is truncated at a maximum value to avoid creating excessively large spheres. This cap also prevents generating multiple spheres at nearly the same position for a single object through multi-view input. The new sphere succeeds the view sample coverage from the old sphere by $S_{\text{new}} = S_1 \cap S_2$.

4 Evaluation

We evaluate our system in terms of usability, task load, and the resultant view synthesis quality from images collected by our approach at different scene scales.

4.1 Study setup

Design We designed a repeated-measures within-subjects study to identify the characteristics of our proposed visualization approaches. We compared the following three approaches (i.e., independent variables in this study):

- `NV`: The video stream is presented without any guidance.
- `SC`: Only spatial coverage is visualized. This is analogous to approaches with spatial analysis.
- `Ours`: Both spatial and angular coverages are visualized.

Metrics To evaluate the system implementations, we collected System Usability Scale (SUS) [14], NASA Task Load Index (NASA-TLX) [8], Satisfaction (Satis.), and the number of collected images. Satisfaction measured evaluations on a 7-point scale between (1) "Totally disagree" to (7) "Totally agree," with the question: "I am satisfied with the scene scanning." per method. Our post-task questionnaires were designed specifically around the view sampling task. Participants ranked the three approaches for each of the following questions: (Q1): "Which interface did you find most comfortable?" (comfort), (Q2): "Which interface did you find the most enjoyable to use?" (enjoyment), (Q3): "Which interface most effectively supported your task?" (task support), (Q4): "Which interface felt the most intuitive and natural to use?" (naturalness), (Q5): "Which interface provided a better sense of spatial awareness?" (spatial awareness), and (Q6): "Which interface did you prefer overall?" (overall performance). We also collected open-ended feedback by asking participants to comment on the following questions: "What was the hardest part about performing the tasks?" and "What do you think can be improved in general?". For evaluating view synthesis quality with the collected image data, we processed the multi-view images and generated Nerfacto [31] and 3DGS [12] scenes. We measured image quality using standard metrics, including peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [32], and LPIPS [37].

Participants We collected 12 participants (two female and 10 male) with an average age of $\bar{X} = 29.2$ (SD$=4.2$) years, all right-handed and with corrected vision. All participants were university students in computer science and scored their AR experience as $\bar{X} = 5.2$ (SD$=1.5$) on a 7-point scale, where 1 represents 'never experienced' and 7 represents 'regular user.' This experiment was approved by the local institutional review board.

Apparatus We used the Apple iPhone 15 Pro. The software was implemented using Unity, C#, and Shader Lab, with Unity ARFoundation for augmented reality features, including device tracking and per-frame depth maps. The captured images are saved asynchronously on the local device every 0.2 seconds, and keyframes are sent to the server every 5 seconds for vision processing. On the server side, we use a desktop PC with an AMD Ryzen 9 7900X 4.7 GHz CPU, 64 GB RAM, and an NVIDIA GeForce RTX 4090 24 GB GPU. Data is communicated between the mobile device and the server using Hypertext Transfer Protocol (HTTP).

Procedure After filling out a consent form, each participant was introduced to a brief training session, which took approximately five minutes. They were taught to hold and move the device in portrait mode. They learned the purpose of view sampling and the way to complete view sampling tasks with each visualization approach. They were instructed to complete the task as quickly and precisely as possible. After the training session, the main session commenced. Participants were invited to one of the three environments (Figure 5), where one of the visualization approaches was

selected and presented. To avoid biases, we employed a 3×3 Latin square design. Participants began the scene capturing by taking an initial image and continued until they felt the sampling was sufficient. After the view sampling session, they answered questionnaires. This procedure was repeated until all visualizations were evaluated. Finally, participants were asked follow-up questions. The entire procedure took approximately one hour.

Dataset We conducted our user study in three scenes (Figure 5). The design of our dataset is analogous to the Shiny dataset [33], which primarily evaluates view-dependent effects using MPI and spherical harmonics-based view synthesis. The Shiny dataset consists of scenes containing specular objects. Similarly, we arranged objects with reflective and transparent materials to emphasize challenging view-dependent phenomena.

In most view synthesis research, a single person collects all view samples for a scene, and these are split into training and evaluation sets. This approach, however, only evaluates performance on views close to the original sampling trajectory and is not suitable for comparing different view sampling strategies, which is our focus. In fact, there is a report that biased ground truth selection can strongly influence final view synthesis results [35]. To address this, we propose using an independent set of ground truth captured by another person (i.e., an examiner) to exhibit broader spatial and angular variance. Because the scenes were open environments with naturally changing lighting conditions, an examiner collected the ground-truth image set immediately after each participant's trial to minimize scene drift between training and evaluation datasets.

All input images were captured at 1920×1440 pixels and resized to 480×360 pixels for vision processing. We used COLMAP [27], a widely adopted structure-from-motion software, to estimate camera poses for both the participants' images and 20 ground truth images. These ground truth images were randomly selected from the full stock to avoid intentional selection and potential biases as much as possible. Using the participants' input data, we optimized Nerfacto and 3DGS of their official implementations[2] and synthesized views at the ground truth viewpoints.

4.2 View synthesis quality

We compared our approach with two baseline approaches and quantitatively evaluated how effective our view sampling strategy is in both scene representations.

Results Table 2 summarize the results of the 3DGS and Nerfacto view synthesis. Our method outperforms the baselines in all image quality metrics. `Ours` achieved the highest performance, followed by `SC` and `NV`. Figures 6 and 7 present qualitative comparisons of 3DGS and Nerfacto view synthesis results among the three methods. Our approach generates high-quality, consistent reconstructions throughout the entire scene, supported by spatial and angular coverage analysis. As demonstrated in the object-centered rendering results, our method effectively captures challenging aspects such as specular reflections and transparency, which baseline methods often miss. These view-dependent effects are most clearly illustrated in the supplemental materials.

Discussion Overall image quality scores are relatively lower ($<$ 20 dB) compared to those typically reported in the literature (23–30 dB). As discussed in the previous section, our view synthesis task is more challenging than the standard benchmarks. However, if we follow the standard practice used in the literature and select ground truth images from among participant-captured images, the scores for all three methods improve significantly by 37%, 15%, and 49% in PSNR, SSIM, and LPIPS, respectively. This improvement reaches an average PSNR of approximately 24 dB, which is in agreement with the expected scores and confirms the difficulty of this new view sampling task and our evaluation setting.

[2]Nerfacto v1.0.1: https://github.com/nerfstudio-project/nerfstudio and 3DGS (Commit: 54c035f): https://github.com/graphdeco-inria/gaussian-splatting

Table 2: Quantitative comparisons against baseline data collection strategies in view synthesis quality.

	3DGS [12]		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)
`NV`	16.338 (3.346)	0.752 (0.142)	0.292 (0.120)
`SC`	17.086 (3.435)	0.800 (0.132)	0.253 (0.112)
`Ours`	**18.891 (3.195)**	**0.848 (0.111)**	**0.201 (0.102)**
	Nerfacto [31]		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)
`NV`	16.115 (2.592)	0.558 (0.088)	0.589 (0.072)
`SC`	16.372 (2.488)	0.574 (0.075)	0.570 (0.069)
`Ours`	**18.134 (1.859)**	**0.616 (0.068)**	**0.520 (0.065)**

4.3 User preferences and comments

Results We performed either ANOVA or Friedmann tests depending on whether the data sphericity and normality were met.

The Friedman test revealed a significant main effect on SUS ($\chi^2(2) = 10.30$, $p = 0.006$, Kendall's $W = 0.43$). Post-hoc pairwise comparisons using Wilcoxon signed-rank tests with Bonferroni correction indicated a significant difference (`NV`: $\bar{X} = 64.38$, SD $= 22.41$; `SC`: $\bar{X} = 83.96$, SD $= 10.97$; $p = 0.02$, $r = 0.74$). The ANOVA revealed a significant main effect on TLX ($F(2,22) = 9.88$, $p < 0.001$, $\eta^2 = 0.17$). Post-hoc pairwise comparisons (Benjamini-Hochberg FDR correction) indicated significant differences between `NV` and `SC` (`NV`: $\bar{X} = 53.42$, SD $= 21.17$; `SC`: $\bar{X} = 30.53$, SD $= 18.77$; $p < 0.01$, Cohen's d $= 1.10$) and between `NV` and `Ours` (`NV`: $\bar{X} = 53.42$, SD $= 21.17$; `Ours`: $\bar{X} = 39.92$, SD $= 21.34$; $p = 0.02$, Cohen's d $= 0.61$).

The Friedman test revealed a significant main effect on Satis ($\chi^2(2) = 15.20$, $p = 0.0005$, Kendall's $W = 0.63$). Post-hoc pairwise comparisons indicated significant differences between `NV` and `SC` (`NV`: $\bar{X} = 4.08$, SD $= 1.44$; `SC`: $\bar{X} = 6.25$, SD $= 0.72$; $p < 0.01$, $r = 0.83$) and also between `NV` and `Ours` (`NV`: $\bar{X} = 6.25$, SD $= 0.72$; `Ours`: $\bar{X} = 6.08$, SD $= 0.95$; $p < 0.01$, $r = 0.83$). The Friedman test revealed a significant main effect on the number of collected images ($\chi^2(2) = 6.50$, $p = 0.04$, Kendall's $W = 0.27$). Post-hoc pairwise comparisons indicated significant difference (`SC`: $\bar{X} = 231.67$, SD $= 131.31$; `Ours`: $\bar{X} = 320.17$, SD $= 169.03$; $p = 0.02$, $r = 0.75$).

Although several effects reached statistical significance, some effect sizes (e.g., Kendall's $W < 0.70$) indicated only moderate agreement. Given the limited sample size, the statistical power for detecting such effects may be insufficient. This limitation should be considered when interpreting the results, and future studies with larger samples are warranted to confirm these findings.

Figure 9-bottom presents the results of the post-task questionnaires. No participants selected `NV` as their top choice for any question. Between `Ours` and `SC`, a noticeable difference emerged for Q5, with `Ours` providing more effective three-dimensional spatial feedback. `SC` was ranked lowest in all questionnaires once, while `Ours` was ranked lowest twice, in Q2 and Q4. Otherwise, no critical differences were observed between the two.

Discussion `SC` received higher SUS scores compared to `NV`. No statistically significant difference was observed for `Ours` in SUS. However, the system performance of `Ours` was well acknowledged in the post-trial evaluations (Figure 9-bottom). We believe that the additional spherical proxies introduced in `Ours` may have contributed to an increased cognitive load for participants, which could explain the slightly lower SUS scores.

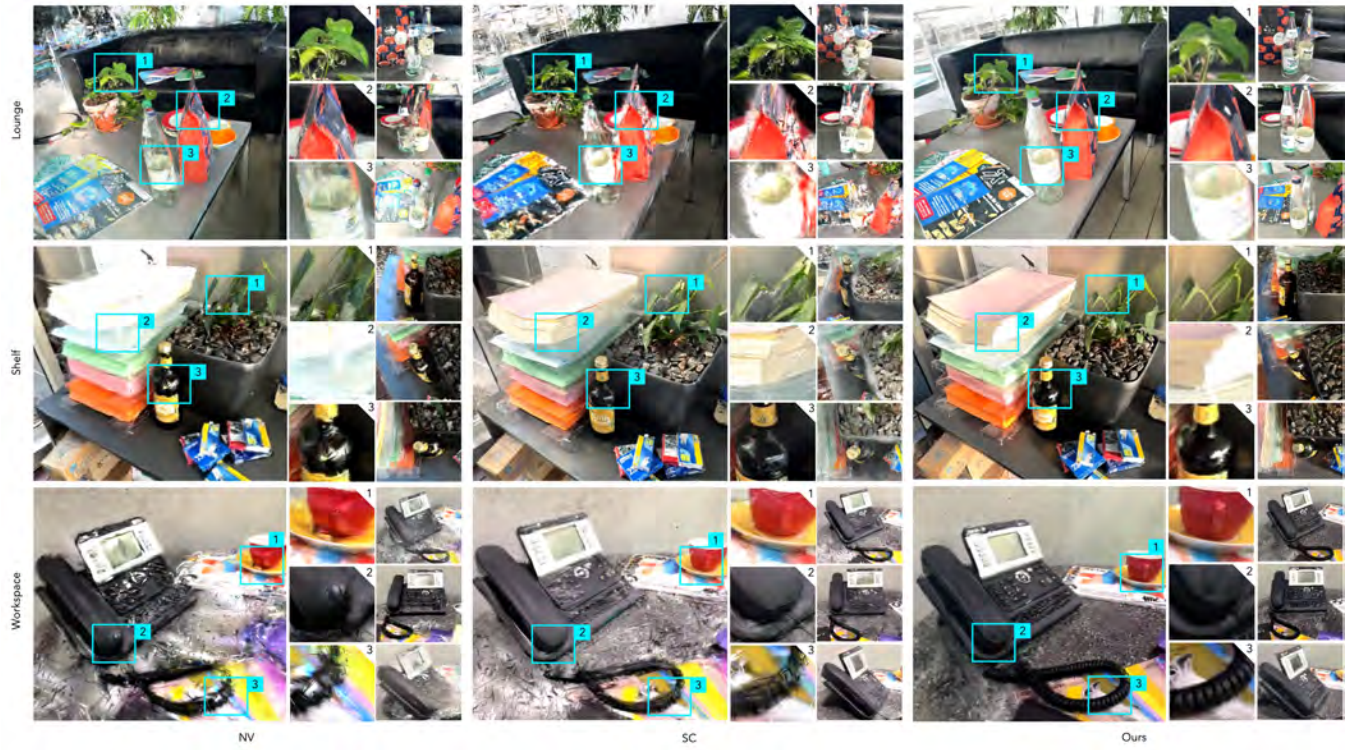

Figure 6: 3DGS [12] view synthesis results using data collected by ours and baseline approaches. The rightmost columns in each approach show how reflective and transparent objects are reproduced at different angles from the collected views. `Ours` exhibits fewer artifacts across different scenes. `NV` suffers from artifacts all around the scenes, and `SC` fails to reproduce detailed and glassy objects.

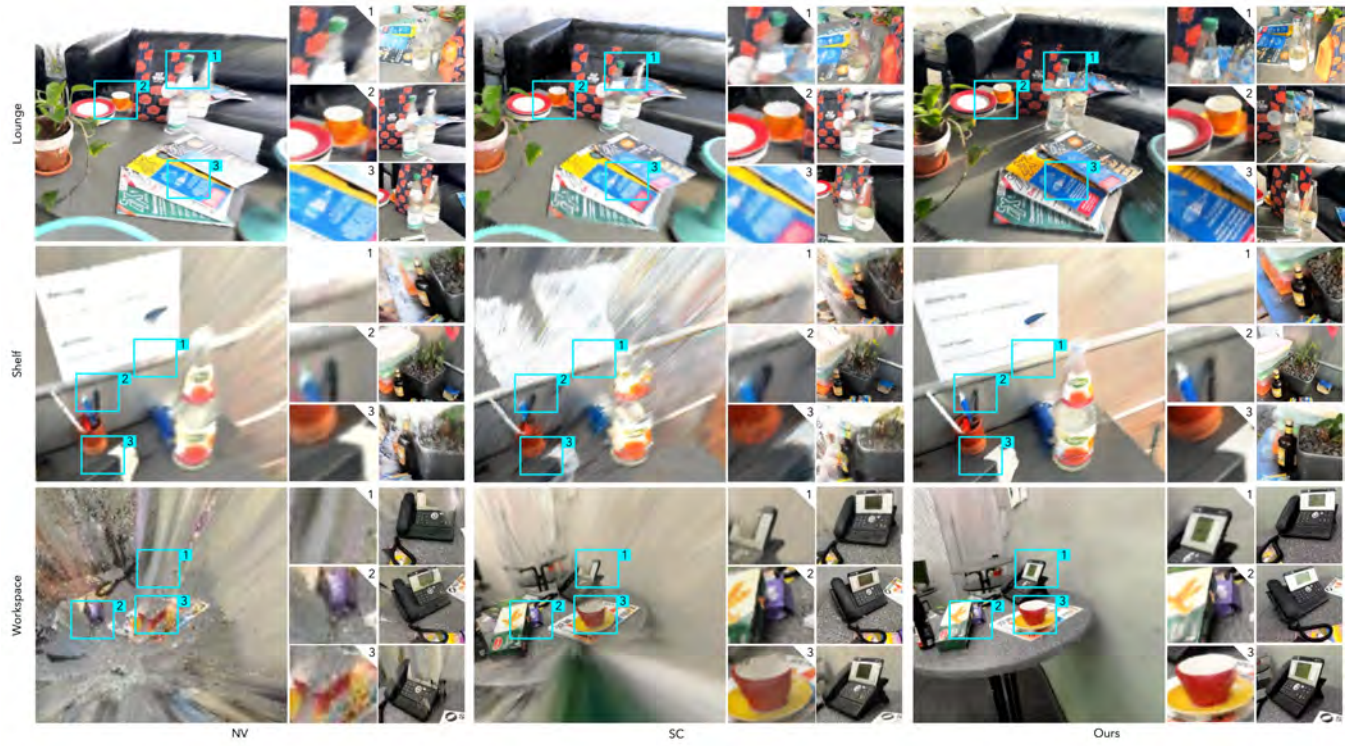

Figure 7: Nerfacto [31] view synthesis results using data collected by ours and baseline approaches. The rightmost columns in each approach show how reflective and transparent objects are reproduced at different angles from the collected views. While `NV` and `SC` show characteristic artifacts of neural rendering, `Ours` provides coherent view synthesis across the test scenes.

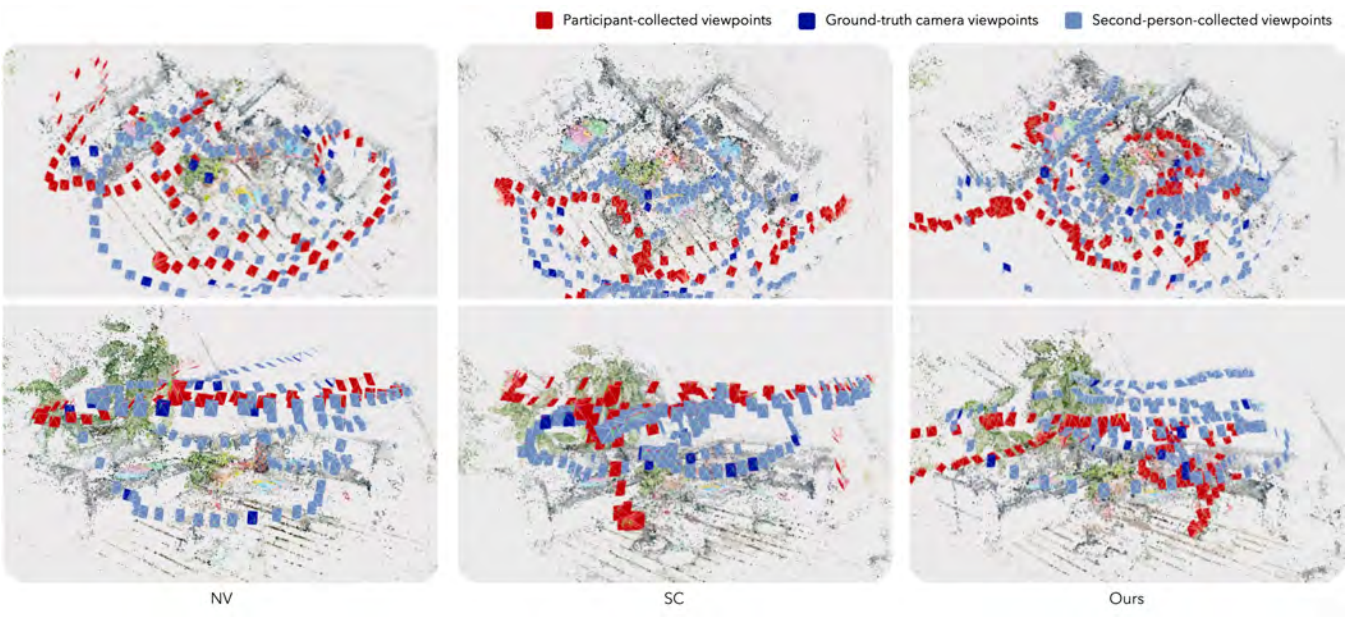

Figure 8: Camera viewpoint visualization in the Lounge scene with NV, SC, and Ours. Red: Training viewpoints collected by the participants; Blue: Ground-truth viewpoints randomly sampled from images collected by a second person to avoid arbitrary evaluation; Light blue: All the other viewpoints collected by a second person.

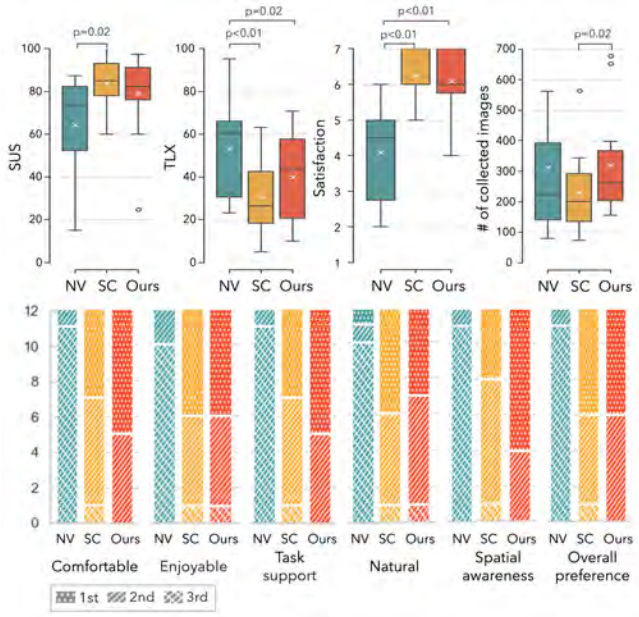

Figure 9: Results of the user study.

Similar trends were observed in NASA-TLX and Satis scores. Both Ours and SC showed significantly lower mental workload and higher satisfaction compared to NV. No significant difference was found between Ours and SC. Since NV lacked any visual guidance, a participant expressed uncertainty or insecurity about whether all parts of the scene had been sufficiently scanned (P8).

Participants using Ours captured more images than those using SC, which may account for the increased perceived workload. Several participants noted challenges related to angular sampling, such as "Getting all the suggested angles is cumbersome" (P5, P6, P8) and "There are some angles that are difficult to reach" (P10). Interestingly, participants using NV and Ours collected a similar number of images, while SC resulted in significantly fewer captures. The fact that Ours achieved the best view synthesis performance (Table 2) suggests that it guided users toward additional viewpoints SC failed to capture. It also demonstrates a more efficient view sampling than NV. Some participants noted that they would have better understood and appreciated the additional effort if they had seen the final rendered results. Four participants (P1, P2, P8, P9) specifically mentioned this, although real-time feedback was not possible due to the time-consuming nature of COLMAP and 3DGS (or Nerfacto) optimization, which can take several hours.

Participants also gave positive feedback on Ours, such as "I had fun making the spheres disappear" (P3), and "Ours allows me to estimate how many angles I got and how good the result may look" (P8). As for feature requests, the most frequently requested improvement was the ability to preview the final rendered result immediately. The second most requested feature was a clearer signal for task completion. Due to the progressive nature of Ours, the participants were free to explore anywhere, while it was roughly restricted to the areas shown in Figure 5. In some parts, the participants were frustrated by unreachable areas (P7, P12).

Figure 8 visualizes typical data of the participants using different approaches. Since the participants only saw the video stream in NV, they had to rely on their intuition to navigate through the environment. As some participants pointed out, they had no clear clues about what they had captured. Thus, the viewpoints appear distributed rather randomly. SC visualizes which areas have already been observed, and the overall viewpoints appear to cover the space more evenly than in NV. However, the viewpoints do not travel in three dimensions, instead staying at the same height. Ours allowed the participants to capture the necessary number of viewpoints depending on the complexity of the 3D structure of the scene. As visual proof, the viewpoints traveled comprehensively within the space, while partly focusing on a cluster of objects on the table.

We calculated the distances between views collected by the participants and those of the ground truth. Since the scale between

point clouds was unknown, we employed Coherent Point Drift (CPD) [21] to match the scales between different structure-from-motion point clouds. We used a known size of a glass bottle to calculate the real-world distances on a metric scale. The average distances to the ground truth viewpoints (mean $\pm$ SD) for `NV`, `SC`, and `Ours` were 0.337 ± 0.173 m, 0.320 ± 0.155 m , and 0.239 ± 0.047 m, respectively, with `Ours` achieving the smallest distance. The average angular differences to the ground truth viewpoints for `NV`, `SC`, and `Ours` were 17.060 ± 8.189 °, 18.540 ± 5.405 °, and 12.392 ± 4.338 °, with `Ours` also achieving the smallest angular difference. The ground truth viewpoints were designed to be diverse by randomly sampling camera positions and orientations. Therefore, the fact that `Ours` exhibited the smallest distance and angular difference to the ground truth suggests that our method effectively captures the scene from a diverse range of viewpoints without bias or missing coverage.

5 Limitations and Future Directions

While IntelliCap demonstrated strong performance in efficient view sampling, it also revealed several unique limitations and opened up interesting directions for future research.

Level of detail control We do not fully utilize the LLM scores. One possible extension would be to adaptively increase the number of sphere proxy subsurfaces based on the scores, providing finer granularity for high-priority objects.

Open vocabulary Our system relies on a closed vocabulary system for lightweight mobile implementation used in the user studies. For more scene-adaptive content validation, future work could incorporate more intelligence into this view sampling task, such as vision-language models or an open vocabulary system, to achieve higher flexibility and recognition performance.

Full mobile implementation We rely on a client-server model to offload vision processing from a mobile device. The most computationally intensive part of our pipeline is Detectron2, which can technically be implemented on mobile devices. We further anticipate that mobile neural processors will bring more mobile-friendly vision processing to enhance view-sampling tasks.

Wide FoV imaging and displaying Due to video stabilization, the current generation of smartphones decreases FoV when AR functionalities are active. Despite this, a larger FoV is advantageous to our task as it allows faster spatial scanning and object identification for both machines and human operators. However, note that different FoV in imaging and displaying would require additional adjustments to our visualization approaches.

Large scale and outdoor scenes While our approach can technically be scaled to larger scenes, our current results are limited to below 10 m^2 to perform the user study in a reasonable time, as well as additional immediate view sampling of the ground truth dataset to conduct view synthesis evaluations. Here, further development in evaluation schemes for multiple users is needed. Applying the method to outdoor scenes would face more severe temporal and weather changes that can lead to inconsistent brightness, shadows, and noises. Dynamics would be handled by identifying and excluding potentially moving objects such as `person`, `car`, and `bicycle` [36] or using sophisticated reconstruction methods [26]. Precise sphere alignment becomes increasingly critical for distant objects, while the generation can be constrained to reachable areas.

Efficient task completion We concluded that potential increases in time and mental workload from additional visualizations and longer photographing sessions would be acceptable to obtain better final view synthesis results. To improve efficiency and user engagement during the view sampling process, multimodal feedback such as visual indicators, audio cues, and haptic responses can help guide and motivate users. These enhancements might include progress bars and rewarding feedback like confirmation sounds or vibrations upon view sampling. However, as discussed in Section 2, defining "sufficient" sampling and task "completion" remains inherently difficult, as these can only be properly evaluated after the final view synthesis.

Misidentification and misalignment Several factors can hinder the effectiveness of our approach. For instance, Detectron2 may occasionally misclassify a simple object as complex or vice versa. Although we did not observe such cases during our study, such misclassifications could lead users to capture an excessive number of images unnecessarily or overlook important objects. Sphere misalignment is another potential issue, which may result in floating spheres that confuse users during the capture process. Our sphere merging strategy is designed to address multiple detections of cluttered, complex objects by grouping them efficiently, thereby enabling the capture process to be completed within a reasonable timeframe. However, in edge cases where the merging policy reaches the maximum allowable sphere size, a smaller adjacent sphere may be generated. This can lead to suboptimal grouping and reduced efficiency in certain scenarios.

Diversity Our visualization approach is grounded in heuristically established techniques and has been positively acknowledged throughout the study. However, there remains significant potential to explore more inclusive methods that accommodate individuals with limited color vision, hearing impairments, mobility challenges, and a variety of smartphone devices in their pockets. We believe this represents a new venue for visualization, human-computer interaction, and accessibility research. Furthermore, when moving at excessively high speeds, motion blur can occur in captured images. To mitigate this motion blur, we believe that measuring scan speed and considering user variability would be effective.

6 Conclusion

This paper presents a novel view sampling approach for high-quality view synthesis, capable of handling multiple scales and complex scenes. Using object semantic classification and LLM to assess the scanning requirements of observed objects, our system enables users to discover more diverse and informative viewpoints, resulting in improved synthesis quality. Unlike conventional best practices that prioritize spatial or angular coverage, our method combines both to enable progressive scanning. The system predicts regions and objects that require denser sampling to support human operators without prior knowledge of scene contents and view synthesis. Experimental results demonstrate that this strategy leads to consistent rendering performance and effective task execution.

Given the novelty of our task design, we introduced new evaluation schemes to assess both the view sampling process and the resulting synthesis quality. Specifically, we designed a user study to collect subjective feedback on our visualization choices by comparing different sampling strategies and developed a follow-up sampling scheme performed by examiners to evaluate view synthesis performance. We plan to release our source code to support future research in this emerging area, particularly contributions aimed at enhancing visual feedback for strategic view sampling coverage and user interaction.

Acknowledgments

This work was supported by the Alexander von Humboldt Foundation funded by the German Federal Ministry of Education and Research, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2120/1 – 390831618, and partly by a grant from JST Support for Pioneering Research Initiated by the Next Generation (# JPMJSP2123)

REFERENCES

[1] C. Birklbauer and O. Bimber. Active guidance for light-field photography on smartphones. *Computers and Graphics (C&G)*, 53(PB):127–135, dec 2015. 2

[2] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 307–318, 2000. 2

[3] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. In *Computer Graphics Forum*, vol. 31, pp. 305–314. Wiley Online Library, 2012. 1, 2, 3, 4

[4] O. Erat, M. Hoell, K. Haubenwallner, C. Pirchheim, and D. Schmalstieg. Real-time view planning for unstructured lumigraph modeling. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 25(11):3063–3072, 2019. 3

[5] L. Fink, D. Rückert, L. Franke, J. Keinert, and M. Stamminger. Livenvs: Neural view synthesis on live rgb-d streams. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH) Asia*, pp. 1–11, 2023. 3

[6] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 20061–20070, 2024. 3

[7] A. Hanson, A. Tu, V. Singla, M. Jayawardhana, M. Zwicker, and T. Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[8] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, vol. 52, pp. 139–183. Elsevier, 1988. 5

[9] R. Ishikawa, H. Saito, D. Kalkofen, and S. Mori. Multi-layer scene representation from composed focal stacks. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 29(11):4719–4729, 2023. doi: 10.1109/TVCG.2023.3320248 2

[10] L. Jin, X. Chen, J. Rückin, and M. Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 11305–11312, 2023. doi: 10.1109/IROS55552.2023.10342226 3

[11] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366, 2024. 2, 3

[12] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics (TOG)*, 42:1 – 14, 2023. 1, 2, 5, 6, 7

[13] G. Kopanas and G. Drettakis. Improving NeRF Quality by Progressive Camera Placement for Free-Viewpoint Navigation. In M. Guthe and T. Grosch, eds., *Vision, Modeling, and Visualization*. The Eurographics Association, 2023. doi: 10.2312/vmv.20231222 3

[14] J. R. Lewis. The system usability scale: past, present, and future. *Int. Journal of Human–Computer Interaction*, 34(7):577–590, 2018. 5

[15] Y. Li, Z. Kuang, T. Li, G. Zhou, S. Zhang, and Z. Yan. Activesplat: High-fidelity scene reconstruction through active gaussian splatting. *arXiv preprint arXiv:2410.21955*, 2024. 2, 3

[16] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison. Gaussian splatting slam. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, 2024. 1

[17] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison. Gaussian Splatting SLAM. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[18] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. on Graphics (TOG)*, 2019. 1, 2

[19] P. Mohr, S. Mori, T. Langlotz, B. H. Thomas, D. Schmalstieg, and D. Kalkofen. *Mixed Reality Light Fields for Interactive Remote Assistance*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2020. 1, 2, 3, 4

[20] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics (TOG)*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127 1

[21] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(12):2262–2275, 2010. 9

[22] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 127–136. Ieee, 2011. 1, 3

[23] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pp. 2320–2327, 2011. doi: 10.1109/ICCV.2011.6126513 3

[24] R. Ng. Fourier slice photography. In *ACM Trans. on Graphics (TOG)*, vol. 24, pp. 735–744. ACM, 2005. 2

[25] E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19:5–11, 1997. 4

[26] N. Schischka, H. Schieber, M. A. Karaoglu, M. Gorgulu, F. Grötzner, A. Ladikos, N. Navab, D. Roth, and B. Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic neural radiance fields. *IEEE Robotics and Automation Letters*, 10(1):548–555, 2025. 9

[27] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[28] E. Sucar, S. Liu, J. Ortiz, and A. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2021. 3

[29] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. OnePose: One-shot object pose estimation without CAD models. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[30] N. Sünderhauf, J. Abou-Chakra, and D. Miller. Density-aware NeRF ensembles: Quantifying predictive uncertainty in neural radiance fields. pp. 9370–9376, 2023. 3

[31] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2023. 2, 5, 6, 7

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing (TIP)*, 13(4):600–612, 2004. 5

[33] S. Wizadwongsa, P. Phongthawee, J. Yenphraphai, and S. Suwajanakorn. NeX: Real-time view synthesis with neural basis expansion. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 8534–8543, 2021. 6

[34] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 1, 4

[35] W. Xiao, R. Santa Cruz, D. Ahmedt-Aristizabal, O. Salvado, C. Fookes, and L. Lebrat. Nerf director: Revisiting view selection in neural volume rendering. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2024. 6

[36] X. Yang, T. Wang, H. Liu, Y. Jin, C. Lang, and Y. Li. Enhancing multimedia applications by removing dynamic objects in neural radiance fields. pp. 2070–2086, 2024. 9

[37] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. 5

[38] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *Proc. Int. Conf. on 3D Vision*, March 2024. 3

[39] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3