SceneDiff: A Benchmark and Method for Multiview Object Change Detection

Yuqun Wu[1] Chih-hao Lin[1] Henry Che[1] Aditi Tiwari[1]
Chuhang Zou[2†] Shenlong Wang[1†] Derek Hoiem[1†]

[1]University of Illinois at Urbana-Champaign [2]Meta

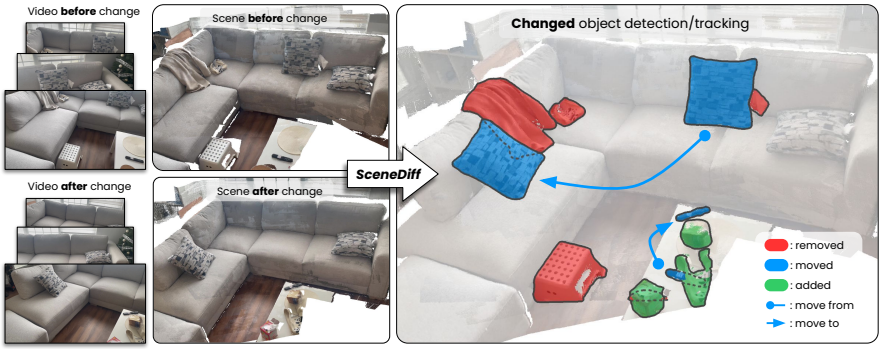

Fig. 1: Multiview change detection. We identify the changed objects (Removed, Added, and Moved) given two videos capturing the same scene at different times. The right panel shows a projected 3D visualization of our 2D predictions, with object boundaries manually overlaid. Dashed lines indicate occluded changed objects.

Abstract. We investigate the problem of identifying objects that have been added, removed, or moved between a pair of captures (images or videos) of the same scene at different times. Accurately identifying verifiable changes is extremely challenging – some objects may appear to be missing because they are occluded or out of frame, while others may appear different due to large viewpoint changes. To study this problem, we introduce the SceneDiff Benchmark, the first multiview change detection dataset for scenes captured along different camera trajectories, comprising 350 diverse video pairs with dense object instance-level annotations. We also introduce the SceneDiff algorithm, a training-free approach that solves for image poses, segments images into objects, and compares them using semantic and geometric features. By building on pretrained models, SceneDiff generalizes across domains without retraining and naturally improves as the underlying models advance. Experiments on multiview and two-view benchmarks demonstrate that our method outperforms existing approaches by large margins (51.6% and 30.6% relative AP improvements). Project page: `https://yuqunw.github.io/SceneDiff`

Keywords: Object-Level Change Detection · Multiview Perception · 3D Scene Understanding · Robotics

† Corresponding authors

1 Introduction

Object change detection (Fig. 1), identifying objects that have been added, removed, or moved based on videos or images of the same scene at different times, serves as a fundamental test of spatial understanding. This capability is critical for applications ranging from robotic room tidying (Fig. 9) to construction monitoring, warehouse inventory verification, and post-disaster damage assessment. For example, a superintendent may want to know where new items have been installed; a warehouse operator needs to confirm that pallets are in the correct locations after a shift change; and disaster responders need to catalog what has been displaced or destroyed. However, the task is extremely challenging: significant viewpoint shifts, lighting variations, and occlusions often cause objects to falsely appear changed. To succeed, a method must establish correspondence between the two sets of frames and identify confirmable changes while ignoring apparent changes that are due only to viewpoint or lighting differences.

In this work, we offer SceneDiff Benchmark, the first instance-level multiview change detection dataset designed for scenes captured along different camera trajectories. The dataset contains 350 real-world sequence pairs from 50 diverse scenes across 20 unique scene categories, consisting of 200 manually collected video pairs and 150 egocentric video pairs extracted from the HD-Epic dataset [32]. Unlike prior benchmarks that assume near-duplicate viewpoints or provide only semantic labels, SceneDiff Benchmark provides dense object instance-level annotations across a large and diverse collection of indoor and outdoor scenes. We provide a SAM2-based [36] annotation tool and an evaluation protocol for both per-view and per-scene object change detection assessment.

Analogous to the "diff" command for text files, our approach, SceneDiff, first aligns scene views and then detects differences. Our two key insights are: (1) feed-forward 3D models can use static scene elements to co-register the before and after sequences into a shared 3D space; and (2) once aligned, changed objects produce geometric and appearance inconsistencies that are easier to detect through object-level region features. By aligning scenes and segmenting them into objects before matching, SceneDiff greatly simplifies comparison and improves accuracy relative to methods that operate at the pixel or point level. We leverage pretrained 3D (e.g., π^3), segmentation (e.g., SAM), and semantic (e.g., DINOv3) foundation models for alignment and comparison, identifying objects with inconsistent appearance or geometry across views. Because SceneDiff is built entirely on pretrained models, it is training-free and straightforward to apply across diverse datasets and domains—handling both image pairs and video sequences—and it naturally improves as the underlying semantic features, 3D representations, or segmentation models advance, as validated in our ablations (Sec. 5.3). This approach outperforms prior work by large margins on both our proposed SceneDiff Benchmark (51.6% relative improvement, averaging obj/im AP in the two portions) and the established two-view RC-3D benchmark [38] (30.6% relative AP improvement). We demonstrate a robotic application (Sec. 5.4), where our method enables a robot to tidy up a messy table to its original, user-defined setup.

In summary, our key contributions are:

- **SceneDiff Benchmark**, the first multiview dataset for object change detection across different camera trajectories, containing 350 real-world sequence pairs from 50 scenes across 20 scene categories, with dense object instance-level annotations spanning both indoor and outdoor environments. We release the dataset, annotation tool, and evaluation code.
- **SceneDiff algorithm**, a training-free framework that co-registers images and segments them into objects before comparison, greatly simplifying matching relative to patch-level approaches. By building entirely on pretrained foundation models, SceneDiff generalizes across domains without retraining and naturally benefits as the underlying models improve. SceneDiff outperforms prior methods on both our proposed benchmark and the established two-view RC-3D benchmark [38], and we demonstrate its practical utility through a closed-loop robotic tidying application.

2 Related Work

Object change detection aims to identify objects in the same scene that change over time. Recent solutions address this problem in terms of two-view [6, 22, 23, 31, 33, 37, 38, 46, 47, 49] and multiview [9, 24, 28, 39, 48] settings. Much effort has been devoted to change detection under similar viewpoints, using learning-based [22, 35] or training-free methods [7, 17]. To address different viewpoints using synthetic training data, CYWS [37] learns the differences at the feature level and performs bounding box detection of changes in the scene through a U-Net architecture. Subsequent work [38] warps image features from one image to another via estimated monocular depth. Other works also train models for dense semantic mask predictions [31]. These methods have demonstrated strong results, though they typically require task-specific training data and can face a sim-to-real gap. Other explorations assume sensor depth inputs and include better feature representations such as neural descriptor fields [9], or intermediate representations such as point cloud [3, 20, 48, 60], 3D semantic scene graph [24], and TSDF [8,34,41,42]. Recent work [10,12,13,25] uses NeRF [27] or 3D Gaussian splatting [16, 58] to identify discrepancies between rendered and captured images. These render-and-compare approaches effectively leverage 3D consistency, though they perform best when dense input views are available and camera trajectories provide good coverage of the scene. In contrast, our training-free approach leverages geometry [54], semantic [44], and segmentation [18] foundation models to robustly detect changes. Furthermore, our proposed benchmark provides a rigorous framework for evaluating these models on the change detection task.

Change detection benchmarks. Existing change detection benchmarks [1, 10, 25, 30, 38, 40] have driven significant progress in the field, primarily focusing on per-view predictions. Datasets in real-world outdoor [1, 40] and synthetic indoor environments [30] established pixel-level metrics under the assumption of similar viewpoints. RC-3D [38] introduces viewpoint variation and provides 100 image pairs, each containing a single changed object. 3DGS-CD [25] and

Table 1: Comparison with existing change detection datasets. The SceneDiff Benchmark is the first dataset with instance-level annotations for video sequences and contains the largest collection of different-viewpoint sequence pairs. *Consistent ID* indicates consistent cross-view instance annotation; *Viewpoints* indicates whether the before/after viewpoints are similar or different; *Out./In.* refer to outdoor/indoor; *Data* denotes the raw video or image count; *# Pairs* denotes the number of frames with ground-truth annotations; *Co-Vis* denotes bidirectional co-visibility between each frame and its best-matching counterpart in the paired sequence, or its given paired frame.

Dataset	Consistent ID	Real	Viewpoints	Annotation	Scene	Data	# Pairs	Co-Vis
VL-CMU-CD [1]	✗	✓	Similar	Semantic	Out.	152 Videos	4.2K	95%
PSCD [40]	✗	✓	Similar	Instance	Out.	770 Images	770	98%
ChangeSim [30]	✗	✗	Similar	Semantic	In.	80 Videos	130K	85%
3DGS-CD [25]	✗	✓	Different	Semantic	In.	5 Videos	20	84%
PASLCD [10]	✗	✓	Different	Semantic	Both	20 Videos	500	85%
RC-3D [38]	✗	✓	Different	Instance	In.	100 Images	100	71%
SceneDiff (Ours)	✓	✓	**Different**	**Instance**	Both	**350 Videos**	6K	65%

PASLCD [10] provide high-quality semantic change maps for 5 and 20 sequence pairs, which are well-suited for testing change detection under test viewpoints interpolated from densely sampled training views. To evaluate a method's ability to consistently identify the same physical changed objects across frames, we offer the first dataset for multiview *object-level* change detection, including *350 video pairs* in diverse real-world scenes (Tab. 1).

Geometry reconstruction plays a critical role in scene change detection, as accurate geometry supports explicit geometric and appearance cues for identifying changes. Traditional methods [19, 21, 26, 43, 56, 57] solve the problem via a two-stage process. More recently, DUSt3R [53] introduces a unified pipeline that directly predicts geometry given two views and shows strong performance. Much effort has been invested to further enhance these pipelines for multiview inputs [15, 45, 51, 54, 55] and dynamic sequences [50, 52, 59]. Unlike these works that focus on dynamic scene reconstruction, we aim to solve the scene change detection problem in static scenes captured at different times. Multiview change detection also serves as a downstream task to evaluate 3D inference models, and we use π^3 [54] for geometry reconstruction in our method.

3 SceneDiff Benchmark

The SceneDiff benchmark evaluates consistent object-level change detection at both per-image and per-scene levels under varying viewpoints (Tab. 1). In this section, we detail the evaluation metrics, dataset statistics, and annotation tool.

3.1 Metrics

We offer three metrics for evaluation in SceneDiff, going from per-pixel and view-centric to per-object and scene-centric: (1) **px/im IoU**: which pixels changed in each image; (2) **obj/im AP**: which objects changed in each image; (3) **obj/sc AP**: which objects changed in each scene. *px/im IoU* and *obj/im AP*

Fig. 2: Sequence Examples. We visualize video pairs before and after changes. Changed objects are color-masked by change type: `Removed`, `Added`, and `Moved`. The background is masked white. The first pair is from *SD-V*, and the second is from *SD-K*.

Fig. 3: Diverse Scene Examples. Annotated frames from the SceneDiff Benchmark.

evaluate per-view segmentation quality and detection coverage, following existing benchmarks [1, 30, 37, 40]. *obj/sc AP* additionally evaluates scene-level object identification consistency across views, requiring the system to determine whether detections across multiple views correspond to the same physical object or to distinct objects. For example, if an added chair is detected in two images within the same sequence, does that count as one changed object or two; and if a chair is moved across sequences, is that a moved object or separate removed and added objects? The superintendent, warehouse operator, or disaster responder is concerned more with objects in the scene than pixels in images, so we consider the third metric our most important, while the other two allow for broader comparison.

Implementation Details. For *obj/im AP*, detections in each view are sorted by confidence and matched greedily to ground-truth masks using mask IoU with a threshold of 0.5. Each ground-truth object can only be matched once, and unmatched predictions are counted as false positives. For *obj/sc AP*, detections across views corresponding to the same object hypothesis are aggregated into a single scene-level prediction, with confidence computed as the mean of per-view prediction confidences. A scene-level prediction is matched to a ground-truth object if the IoU aggregated across frames in both sequences exceeds 0.5. More results under different criteria are provided in the supplement.

3.2 Dataset

SceneDiff (Fig. 2) contains 350 video sequence pairs and 1009 annotated objects across the *Varied* and *Kitchen* subsets. The *Varied* subset (*SD-V*) contains 200

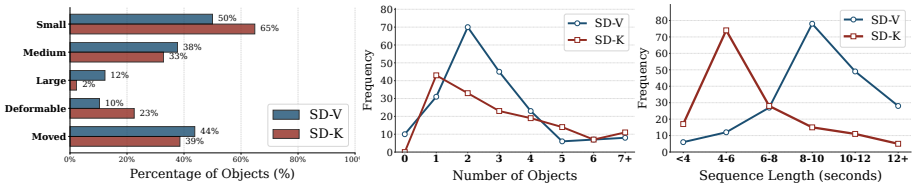

Fig. 4: Dataset Statistics. Distribution of object properties, changed object counts, and sequence lengths in the SceneDiff Benchmark. Object size categorization is based on the average pixel size across all frames. *SD-V* contains larger objects and longer sequences, while *SD-K* contains more deformable objects.

sequence pairs collected by 7 people in a wide variety of daily indoor and outdoor scenes (Fig. 3), and the *Kitchen* subset (*SD-K*) contains 150 sequence pairs from the HD-Epic dataset [32] with changes that naturally occur during cooking activities. For each video pair, we record all changed object attributes, including object names, change types (`Added`, `Removed`, or `Moved`), and deformability, and annotate their full segmentation masks in all visible frames. In total, the sequence collection, tool development, and annotation required approximately 160, 30, and 60 hours respectively. To avoid ambiguity, we avoid capturing objects that are moving within a single sequence, so that all changes are across sequences. An overview of the dataset characteristics is presented in Fig. 4, including the distribution of changed objects per video, duration of video sequences, and object attributes (size, deformability, and category). Our validation set, intended for training or hyperparameter tuning, is 50 video pairs per subset, and the test set is the remaining 250 pairs, with all pairs from the same physical scene placed in the same split.

3.3 Annotation Tool

Annotating dense object masks across video pairs is time-consuming, even with modern tools such as SAM2 [36]. To streamline this process, we develop an annotation interface based on SAM2 in which users upload video pairs, specify object attributes (deformability, change type, multiplicity), and provide sparse point prompts on selected frames via clicking. The system records these prompts, propagates masks throughout both videos offline, and provides a review interface where users can visualize the annotated videos, refine annotations if needed, and submit verified pairs to the dataset. This reduces annotation time from approximately 30 minutes to 10 minutes per video pair. The annotation video and other details are in the supplement.

4 SceneDiff Method

SceneDiff aims to detect the changed objects given two image sequences taken before and after the scene changes (Fig. 5).

Task Definition: Given two videos, $\mathcal{I}_{\text{pre}} = \{\mathbf{I}^n_{\text{pre}}\}_{n=1}^{N_{\text{pre}}}$ and $\mathcal{I}_{\text{post}} = \{\mathbf{I}^n_{\text{post}}\}_{n=1}^{N_{\text{post}}}$, captured before and after the change, our goal is to identify verifiable object-level changes across all views, while ignoring apparent differences caused solely by

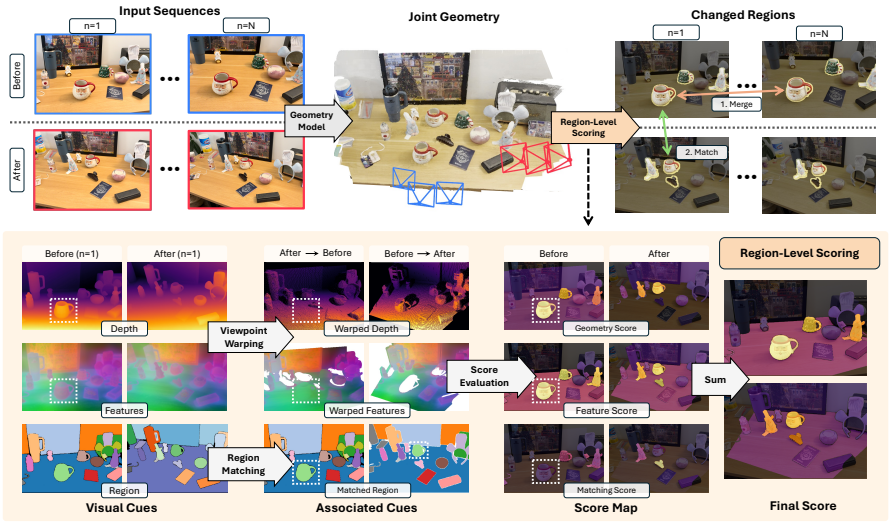

Fig. 5: SceneDiff Method. *Top (Overall Approach):* Given before and after sequences, SceneDiff jointly regresses geometry, selects paired views with high co-visibility, and computes region-level change scores. Changed regions are then merged within each sequence into object-level changes and matched across sequences to classify the change type. *Bottom (Region-Level Change Scoring):* For each paired view, we compute three scores: a geometry score via depth reprojection, an appearance score via feature reprojection, and a region matching score by comparing mean-pooled features across images. These are combined and pooled to produce unified region-level change scores.

viewpoint variations, e.g., occlusion, partial visibility, and different illumination. The output is a set of changed objects with corresponding 2D masks in all visible views and their change type: *Added* (present only in $\mathcal{I}_{\text{post}}$), *Removed* (present only in $\mathcal{I}_{\text{pre}}$), or *Moved* (present in both but at different positions). When $N_{\text{pre}} = N_{\text{post}} = 1$, the task reduces to two-view change detection.

Overall Approach: Scene changes produce geometric and appearance discrepancies in 3D space. However, viewpoint variations can also create differences, making these cues noisy. To address this, SceneDiff first selects view pairs with high co-visibility from the before and after videos to ensure comparable perspectives. It then computes region-level change scores from geometry and appearance cues for each view pair, detects changed regions, associates these regions across time to merge them into consistent object-level changes, and finally produces object-level change segmentation (2D and 3D) and classification.

4.1 Geometry Regression and Frame Pairing

We process the video pair—$\mathcal{I}_{\text{pre}}$ and $\mathcal{I}_{\text{post}}$ jointly through π^3 [54] to estimate depth, pose, and intrinsics, yielding $\{\mathbf{D}^n_{\text{pre}}, \mathbf{T}^n_{\text{pre}}, \mathbf{K}^n_{\text{pre}}\}$ and $\{\mathbf{D}^n_{\text{post}}, \mathbf{T}^n_{\text{post}}, \mathbf{K}^n_{\text{post}}\}$, where $\{\cdot\}$ denotes a set over frame index n. To ensure a consistent scale across scenes, we normalize $\{\mathbf{D}^n_{\text{pre}}, \mathbf{D}^n_{\text{post}}\}$ and $\{\mathbf{T}^n_{\text{pre}}, \mathbf{T}^n_{\text{post}}\}$ so that the reconstructed point clouds fit within a unit cube $[-1, 1]^3$.

We then select frames across "before" and "after" sequences for change detection. Not all frame pairs are suitable for matching due to camera motion and limited overlap. Therefore, for each frame n in the before sequence, we select one or more frames n' in the after sequence, considering $(\mathbf{I}^{n}_{\text{pre}}, \mathbf{I}^{n'}_{\text{post}})$ a good pair if (1) their *bidirectional co-visibility*—the average fraction of mutually visible pixels under before $\leftrightarrow$ after reprojection—exceeds 50%, or (2) $\mathbf{I}^{n'}_{\text{post}}$ has the highest co-visibility among all frames in $\mathcal{I}_{\text{post}}$. The selected pairs are then used to compute change scores; for simplicity, we denote a paired view as $(\mathbf{I}_{\text{pre}}, \mathbf{I}_{\text{post}})$.

4.2 Region-Level Change Scoring

Reprojected depth and appearance differences may reveal scene changes, but two challenges exist: 1) differences may arise from viewpoint variation, e.g., occlusion, rather than actual scene changes; 2) pixel-level discrepancies contain noise and cannot capture object-level changes. To address these issues, we use reprojected depth to distinguish true scene changes from viewpoint-dependent occlusions, leverage DINOv3 [44] features for robust appearance comparison, and aggregate pixel-level cues into region-level scores using masks predicted by SAM [18].

Geometry Comparison: When reprojecting pixels from $\mathbf{I}_{\text{pre}}$ to $\mathbf{I}_{\text{post}}$, objects that **exist only in** $\mathbf{I}_{\text{pre}}$ yield a reliable cue: the observed depth value in $\mathbf{I}_{\text{post}}$ (background) is larger than that of reprojected depth (object), producing positive differences. In contrast, negative differences are ambiguous, as they can arise from occlusion or from objects that **exist only in** $\mathbf{I}_{\text{post}}$. Accordingly, we adopt an *asymmetric* rule: we only use pixels in $\mathbf{I}_{\text{pre}}$ with non-negative depth differences when reprojected to $\mathbf{I}_{\text{post}}$ to detect changes visible in $\mathbf{I}_{\text{pre}}$; by reversing the view order, we detect changes visible in $\mathbf{I}_{\text{post}}$.

Specifically, for a pixel p with coordinates (x_p, y_p) in $\mathbf{I}_{\text{pre}}$, we first transform it into the camera space of $\mathbf{I}_{\text{post}}$ as $\mathbf{p}^{3d}_{\text{post}} = \mathbf{T}^{-1}_{\text{post}}\mathbf{T}_{\text{pre}}\,\mathbf{D}_{\text{pre}}(p)\mathbf{K}^{-1}_{\text{pre}}[x_p, y_p, 1]^{\top}$, and compute the reprojected pixel $p' = \pi(\mathbf{K}_{\text{post}}\mathbf{p}^{3d}_{\text{post}})$ and the corresponding reprojected depth $\mathbf{D}^{*}_{\text{post}}(p') = [\mathbf{p}^{3d}_{\text{post}}]_z$, where $\pi(\cdot)$ denotes projection to image space and $[\cdot]_z$ denotes the Z-component. We then define the depth-difference score $\mathbf{E}_{\text{geom}}$:

$$\mathbf{E}_{\text{geom}}(p) = \mathbf{D}_{\text{post}}(p') - \mathbf{D}^{*}_{\text{post}}(p'). \tag{1}$$

As described, positive values indicate objects only appear in $\mathbf{I}_{\text{pre}}$; negative values reflect occlusion or objects only appear in $\mathbf{I}_{\text{post}}$. To enforce the asymmetric rule, we construct a directional visibility mask that accounts for non-negative differences and fields of view: $\mathbf{M}_{\text{pre}} = \big(\mathbf{E}_{\text{geom}} \geq \tau_{\text{occ}}\big) \wedge \mathbf{V}_{\text{pre}\to\text{post}}$, where $\tau_{\text{occ}} = -0.02$ and $\mathbf{V}_{\text{pre}\to\text{post}}$ is the visibility mask from the pre$\to$post reprojection. We swap the view order to obtain $\mathbf{M}_{\text{post}}$ for detecting changes visible in $\mathbf{I}_{\text{post}}$.

Appearance Comparison: To detect appearance changes, we extract DINOv3 [44] features ($\mathbf{F}_{\text{pre}}$, $\mathbf{F}_{\text{post}}$) and measure feature dissimilarity via reprojection. We then acquire the reprojected appearance features $\mathbf{F}^{*}_{\text{pre}}$ from $\mathbf{F}_{\text{post}}$, apply the directional visibility mask $\mathbf{M}_{\text{pre}}$ to exclude the invisible areas, and obtain the reprojected feature score $\mathbf{E}_{\text{feat}}$. Specifically, given pixel p in $\mathbf{I}_{\text{pre}}$ and the

corresponding pixel p' in $\mathbf{I}_{\text{post}}$:

$$\mathbf{E}_{\text{feat}}(p) = \mathbf{M}_{\text{pre}}(p)\Big(1 - \cos(\mathbf{F}_{\text{pre}}(p), \mathbf{F}^*_{\text{pre}}(p))\Big), \tag{2}$$

where $\mathbf{F}^*_{\text{pre}}(p) = \mathbf{F}_{\text{post}}(p')$ is the reprojected feature of p.

Region-Level Matching: In addition to $\mathbf{E}_{\text{feat}}$, which compares reprojected features pixelwise and then pools over regions, we propose region-level matching, that first pools features and then compares regions at any position. We extract regions ($\mathcal{R}_{\text{pre}}$, $\mathcal{R}_{\text{post}}$) from SAM [18], where each region $r \in \mathcal{R}$ corresponds to a set of pixels from a predicted mask. Reprojection-based cues ($\mathbf{E}_{\text{geom}}$ and $\mathbf{E}_{\text{feat}}$) heavily rely on accurate geometry. To enhance robustness, we additionally measure whether regions have corresponding objects in the other view based on appearance alone. We aggregate features $\mathbf{F}_{\text{pre}}$ within each region defined by masks $\mathcal{R}_{\text{pre}}$ and compute the mean-pooled feature vector $\mathbf{F}^r_{\text{pre}} = \frac{1}{|r|}\sum_{p\in r}\mathbf{F}_{\text{pre}}(p)$ for each region r in $\mathbf{I}_{\text{pre}}$. We then use cosine similarity to select the best-matching region $\sigma(r)$ in $\mathbf{I}_{\mathbf{post}}$ and compute the region matching score E_{region} over r as:

$$E_{\text{region}}(r) = 1 - \cos(\mathbf{F}^r_{\text{pre}}, \mathbf{F}^{\sigma(r)}_{\text{post}}), \tag{3}$$

where $\sigma(r) = \arg\max_{s\in\mathcal{R}_{\text{post}}} \cos(\mathbf{F}^r_{\text{pre}}, \mathbf{F}^s_{\text{post}})$. To account for occlusion and visibility, we exclude regions with more than 50% pixels masked by $\mathbf{M}_{\text{pre}}$ from this matching process.

Score Aggregation: All cost maps are mean-pooled with the region masks, and combined with a weighted sum to obtain the unified score map Δ_{pre} over each $r \in \mathcal{R}_{\text{pre}}$:

$$\Delta_{\text{pre}}(r) = \frac{\lambda^{geom}}{|r|}\sum_{p\in r}\mathbf{E}_{\text{geom}}(p) + \frac{\lambda^{feat}}{|r|}\sum_{p\in r}\mathbf{E}_{\text{feat}}(p) + \lambda^{region}E_{\text{region}}(r) \tag{4}$$

where $|r|$ is the number of pixels in region r and $\lambda^{geom} = 1.0$, $\lambda^{feat} = 0.5$, and $\lambda^{region} = 0.1$ are score weights. Similarly, we compute Δ_{post} by swapping $\mathbf{I}_{\text{pre}}$ and $\mathbf{I}_{\text{post}}$ for all regions in $\mathcal{R}_{\text{post}}$.

4.3 Instance Association and Change Classification

We use the change scores to retrieve the changed regions in each frame. However, the detected per-frame regions across different frames can correspond to the same object. Practically, we often want to know how many object instances changed, not how many regions changed. This requires identifying which regions across frames correspond to the same object instance.

Frame-Level Change Detection: For each view $\mathbf{I}^n_{\text{pre}}$, we compute the averaged unified score maps $\bar{\Delta}^n_{\text{pre}}$ across all its matched frame pairs. For 3D consistency, we unproject all averaged score maps $\bar{\Delta}^n_{\text{pre}}, \forall n$ into 3D, voxelize the resulting point clouds, average scores within each voxel, and mean-pool again for region-level score maps. Given updated $\{\bar{\Delta}^n_{\text{pre}}\}$ and $\{\bar{\Delta}^n_{\text{post}}\}$, we apply a threshold τ_Δ, and

retrieve a set of changed regions $\mathcal{R}^{\triangle}_{\text{before}} = \bigcup_{n=1}^{N_{\text{pre}}} \{r \in \mathcal{R}^n_{\text{before}} \mid \bar{\Delta}^n_{\text{pre}}(r) > \tau_\Delta\}$, with each region r associated with a change score $\bar{\Delta}^n_{\text{pre}}(r)$. To choose the matching threshold τ_Δ, we use the maximum entropy thresholding algorithm [14], which works as well as picking the single best threshold on the validation set, but applies more easily to other datasets.

Video-Level Region Association: To obtain instance-level changes, we merge regions across frames using an iterative procedure inspired by ConceptGraph [11]. A region is considered another view of an existing object if their features and 3D points are similar. Specifically, we initialize our object set $\mathcal{O}_{\text{pre}}$ with $\mathcal{R}^{0,\triangle}_{\text{pre}}$; then for each frame n, we iteratively measure the similarity score between each region $r \in \mathcal{R}^{n,\triangle}_{\text{pre}}$ and the changed objects $o \in \mathcal{O}_{\text{pre}}$:

$$S(r) = \max_{o \in \mathcal{O}_{\text{pre}}} \left(S_{\text{feat}}(o, r) + S_{\text{geo}}(o, r) \right), \tag{5}$$

where $S_{\text{feat}}(o, r) = \cos(\mathbf{F}^r_{\text{pre}}, \mathbf{F}^o_{\text{pre}})$ measures appearance feature similarity between the region and object; and $S_{\text{geo}}(o, r) = \sum_{\mathbf{x} \in \mathcal{P}^r_{\text{pre}}} \mathbf{1}\left(\min_{\mathbf{y} \in \mathcal{P}^o_{\text{pre}}} \|\mathbf{x} - \mathbf{y}\|_2^2 < \sigma_{\text{geo}} \right) / |\mathcal{P}^r_{\text{pre}}|$ measures region-to-object geometry similarity as the fraction of region's points close enough to the object's point cloud with $\sigma_{\text{geo}} = 0.02$. If $S(r)$ is higher than the merging threshold ($\sigma_{\text{merge}} = 1.4$), we merge region r into the most similar object o, updating the object's feature via a running average and its point cloud via concatenation:

$$\mathbf{F}^o_{\text{pre}} \leftarrow w^o \, \mathbf{F}^r_{\text{pre}} + (1 - w^o) \, \mathbf{F}^o_{\text{pre}}, \mathcal{P}^o_{\text{pre}} \leftarrow \mathcal{P}^o_{\text{pre}} \cup \mathcal{P}^r_{\text{pre}} \tag{6}$$

where $w^o = 1/(N^o + 1)$ and N^o is the number of regions merged into o. Otherwise, we instantiate a new object o' with $\mathbf{F}^{o'}_{\text{pre}} \leftarrow \mathbf{F}^r_{\text{pre}}$ and $\mathcal{P}^{o'}_{\text{pre}} \leftarrow \mathcal{P}^r_{\text{pre}}$. This process provides a set of changed objects $\mathcal{O}_{\text{pre}}$. We compute the changed objects $\mathcal{O}_{\text{post}}$ in the second video following the same process.

Object Status: We perform one-to-one greedy matching between objects in $\mathcal{I}_{\text{pre}}$ and $\mathcal{I}_{\text{post}}$. Pairs with feature cosine similarity greater than $\tau_{\text{sim}} = 0.7$ are sorted in descending order and matched sequentially. Matched objects are labeled as *Moved* and unmatched objects are labeled as *Added* (if in $\mathcal{I}_{\text{post}}$) or *Removed* (if in $\mathcal{I}_{\text{pre}}$).

Parameters: Feature weight and threshold parameters are set to the indicated values based on the SceneDiff validation set, with the same parameters used across all benchmarks. The supplemental includes sensitivity analysis of thresholds.

4.4 Two-Image Input Case

There may be only one before image and one after image, instead of two videos. In that simpler case, we skip 3D consistency aggregation and video-level region association, treating each detected region as an independent object and applying the change type classification directly at the region level with the same parameters.

Table 2: Multiview Change Detection on SceneDiff test set. We report metrics that correspond to each method's output type: for pixel masks, px/im IoU; for bounding boxes, obj/im AP; for object-level multiview tracks, obj/sc AP. Bounding boxes are converted to masks using SAM for mask-based obj/im AP. Object-level tracks are converted to per-image instance masks or pixel masks for comparison. * indicates results after finetuning on our validation set.

	Output Type	**Pixel**	**Bounding Box**			**Object Region Tracks**		
	Method	MV3DCD	CYWS-3D	CYWS-2D	CYWS-2D*	VLM	3DGS-CD	**SceneDiff**
SD-V	*px/im IoU*	22.1	-	-	-	22.9	14.7	**39.1**
	obj/im AP	-	13.9	19.4	24.5	7.1	1.0	**43.8**
	obj/sc AP	-	-	-	-	5.3	0.5	**22.8**
SD-K	*px/im IoU*	9.1	-	-	-	12.4	4.8	**20.8**
	obj/im AP	-	6.6	9.0	16.8	3.6	0.1	**20.9**
	obj/sc AP	-	-	-	-	2.1	0.1	**10.6**

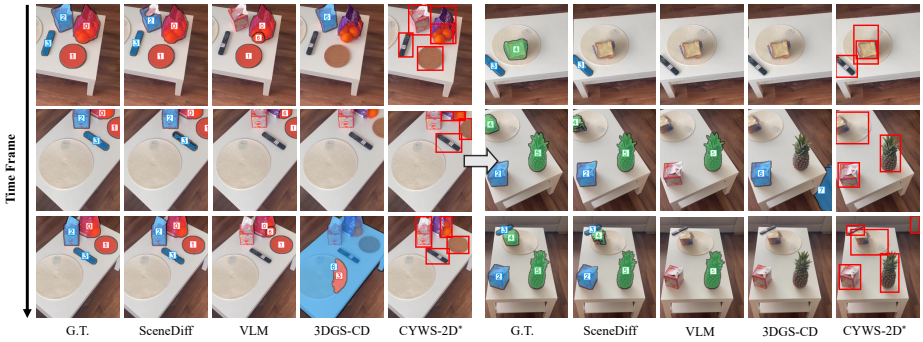

Fig. 6: Qualitative comparison on the SceneDiff benchmark. G.T. objects are visualized with numbers. Object-level predictions show matched GT IDs, or unique IDs if unmatched. Our method correctly predicts all changed objects but misses the bread in a single view. 3DGS-CD produces some correct per-view detections but struggles to associate them into consistent objects. The VLM baseline names accurate but incomplete changed objects (*"Removed: orange, snack bag, coaster; Added: pineapple, sandwich"*). Finetuned CYWS-2D successfully detects most changes (showing top-5 predictions after NMS). Color map: Removed, Added, and Moved.

5 Experiments

We show results on our new SceneDiff benchmark (Sec. 5.1) and on the two-view change detection dataset (Sec. 5.2). In Sec. 5.3, we ablate key design choices and the impact of using different features and geometry estimation models. We also demonstrate a robotic application in Sec. 5.4. More results, e.g., failure cases, robustness of geometry models under change, predictions with dynamic contents, time analysis, robustness with different hyperparameters, are provided in the supplement.

5.1 Two-Sequence Change Detection

We present the evaluation results in Tab. 2, comparing SceneDiff against pair-wise detection methods [37, 38], 3D Gaussian Splatting (3DGS) based approaches [10,

Fig. 7: A more challenging sequence pair. Color map: Removed, Added, and Moved.

Table 3: Average inference time for a 10-second sequence pair on an A40 GPU.

Method	SceneDiff	VLM	CYWS-2D	CYWS-3D	3DGS-CD	MV3DCD
Time	159.4s	30.5s	18.4s	23.9s	>30mins	>30mins

25], and a vision-language model (VLM) baseline. SceneDiff outperforms all baselines across all metrics, especially in *obj/sc AP*. However, there is much room for improvement, particularly for the *SD-K* subset, which features complex and cluttered scenes from an egocentric viewpoint. For pair-wise detection methods CYWS-2D and CYWS-3D [37, 38], we apply our view-pairing step (Sec. 4.1) and convert box predictions to masks using SAM [18]. We finetune CYWS-2D on 3125 frame pairs from the validation set and use the official pretrained model for CYWS-3D (training code not released). 3DGS based methods [10, 25] rely on accurate novel view rendering for change detection. However, high-quality rendering is challenging for current 3DGS methods given large viewpoint changes across trajectories, even when we use our regressed camera parameters as input and sample more densely (3 FPS) for 3DGS optimization. We follow 3DGS-CD [25] by using SAM to assign the symmetric change prediction from MV3DCD [10] to the correct sequence.

We also evaluate SceneDiff on PASLCD [10], outperforming the best baseline by 1.2 mIoU points (Tab. 4). For the VLM baseline, we use Qwen2.5-VL [2] to predict changed object names and types from both sequences, followed by SAM3 [4] for localization. For AP computation, we use SAM3's mask confidence as the detection confidence, since VLMs generate probabilities that are not calibrated for instance-level ranking. To isolate the VLM's recognition capability from localization errors, we manually match predicted text outputs to ground-truth labels: the VLM correctly names 113/369 changed objects in SD-V (1837 total predictions) and 42/327 in SD-K (792 total predictions), indicating bottlenecks in both recognition and localization. Fig. 6 shows the qualitative comparisons, and Fig. 7 highlights a challenging case, illustrating that this task remains far from solved. See supplemental material for more details and analysis.

Table 4: Change Detection on PASLCD [10]. We report mIoU for pixel-level symmetric change prediction across indoor and outdoor environments. Despite not being the primary focus, SceneDiff generalizes well and achieves competitive results. Baselines are sourced from MV3DCD [10]. See full results in Supp.

Method	Mean ↑	Indoor ↑	Outdoor ↑
CSCDNet [40]	12.5	11.7	13.4
CYWS-2D [37]	27.3	32.9	21.7
MV3DCD [10]	46.1	44.9	47.4
SceneDiff	**47.3**	**46.5**	**48.0**

Table 5: Two-View Change Detection on RC-3D [38]. We report AP50 for box predictions across *Present*, *Absent*, and *Both* views. *Blank entries denote metrics unavailable in the original paper or public codebase.* CYWS-2D/3D baselines are sourced from the original paper. * denotes results incorporate sensor depth.

Method	Both ↑	Present ↑	Absent ↑
CYWS-2D [37]	14.0	-	-
CYWS-3D [38]	41.0	-	-
CYWS-3D* [38]	50.0	-	-
SceneDiff	**65.3**	**72.3**	**59.1**

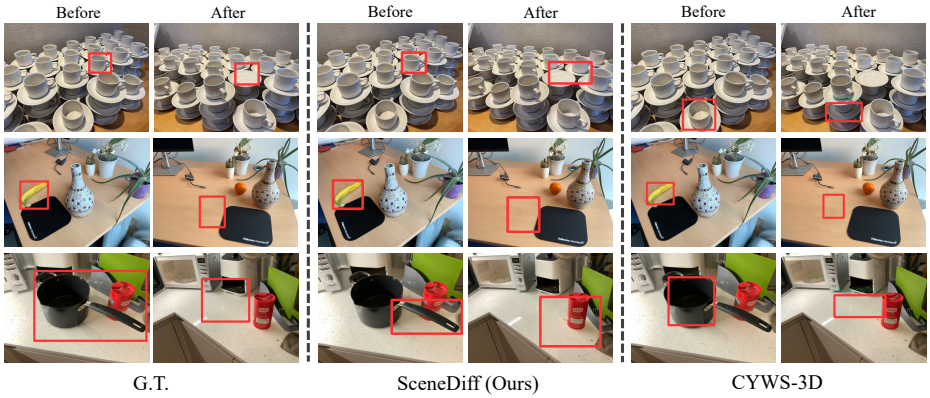

Fig. 8: Qualitative Comparison on RC-3D [38]. We visualize the highest-confidence detections for a single removed object per image pair for Ours and CYWS-3D [38]. SceneDiff relies on SAM-generated masks for prediction, and can misalign with ground truth annotations (3rd row).

Runtime. Tab. 3 reports the average inference time per sequence pair. SceneDiff runs in 159s on a single A40 GPU, far faster than 3DGS-based methods (>30min), and reduces to 65s with batched bf16 inference and caching disabled. Peak GPU memory is 19.2GB. While not real-time, downstream tasks like robotic tidying (Sec. 5.4) compute change detection only once before planning.

5.2 Two-View Change Detection

We compare our method with existing methods on RC-3D [38] in Tab. 5, and show that SceneDiff outperforms all other methods. Qualitative results are provided in Fig. 8. Although pixelwise change detection under similar viewpoints is not our primary target scenario, SceneDiff still outperforms the best prior work [17] by 2.2 points in F1 score on ChangeSim [30] (see the supplement for the full table).

Table 6: Ablation Study on SD-V test set. $E_{\{g,f,r\}}$ denote $E_{\{\text{geom,feat,region}\}}$. Values show the difference (Δ) from the full method (π^3+DINOv3).

Metric	Backbone					Feature Cues (π^3+DINOv3)					
	π^3 +DINOv3	π^3 +DINOv2	π^3 +DINOv1	VGGT +DINOv3	FASt3R +DINOv3	E_g	E_f	E_r	E_g+E_f	E_g+E_r	E_f+E_r
px/im IoU	39.1	+0.4	-3.0	-2.2	-23.7	**+2.1**	-6.1	-14.8	-1.8	-1.0	-4.5
obj/im AP	43.8	+1.3	**+1.5**	-8.1	-39.8	-12.1	+0.1	-14.2	-0.1	-6.9	+0.9
obj/sc AP	22.8	-1.5	-2.7	-8.7	-22.4	-10.3	0.0	-0.8	+0.1	-5.0	**+2.6**

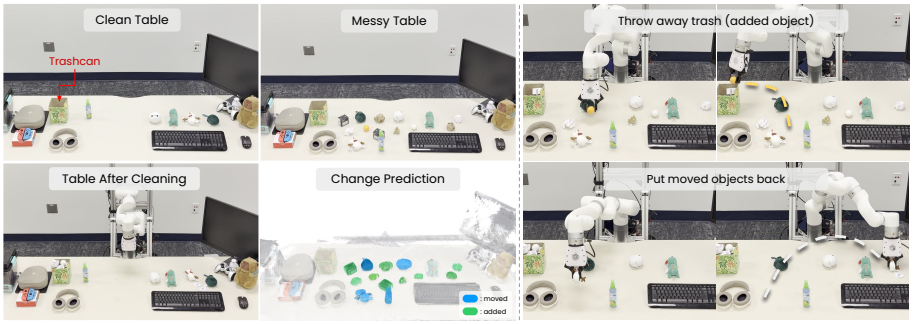

Fig. 9: Real-World Cleaning Robot. The left panel shows our initial "clean" and "messy" table state, and the right shows the robot's execution.

5.3 Ablation Studies

We conduct experiments to evaluate our key design choices and the impact of different 3D estimation models [51, 54, 55] and appearance feature extractors [5, 29, 44] in Tab. 6. Our experiments show that the geometry model really matters for alignment, as the reprojected appearance cue (E_{feat}) is the most important, significantly outperforming the region-level appearance matching cue (E_{region}) that does not require geometry reprojection. The geometry-based change features (E_{geom}) alone are effective for pixel segmentation but not very accurate for scene-level change detection. This may be due to the features being overly sensitive to geometric errors and incapable of finding changes in flat or very small objects. DINO features [5, 29, 44] achieve similar performance, with DINOv3 performing the best for object-level evaluation, demonstrating stronger capacity for recognizing the same objects across different viewpoints and locations.

5.4 Applications

SceneDiff enables various downstream tasks, especially in robotic manipulation. In Fig. 9, we show a demonstration of a robot cleaning a messy table back to the initial clean state by applying SceneDiff to detect moved and added objects. Please refer to the supplement for the video, additional details, and visualizations.

6 Conclusion and Limitations

We presented the SceneDiff Benchmark, the first multiview dataset for object-level change detection across different camera trajectories, with 350 video pairs

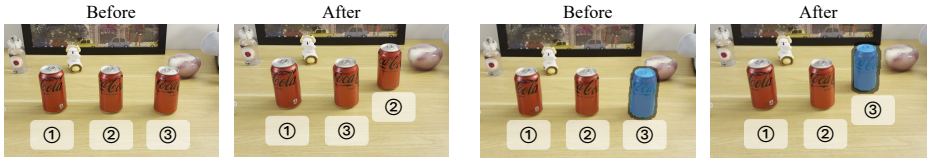

Fig. 10: Ambiguous Case. When the second can is moved elsewhere and the third can is moved into its place, the observations are indistinguishable from only the third can having moved. SceneDiff therefore predicts only the third can as moved.

and dense instance-level annotations. Our training-free SceneDiff algorithm co-registers scenes using pretrained 3D, segmentation, and appearance models, then detects changes through geometric and semantic inconsistencies at the object level—generalizing across domains without retraining and naturally improving as foundation models advance. Experiments show large gains over prior work (51.6% and 30.6% relative AP on our benchmark and RC-3D, respectively), and we demonstrate practical utility through a closed-loop robotic tidying application. The benchmark, annotation tool, and code will be publicly released.

Limitations. Some change configurations are inherently ambiguous from geometry and appearance alone, e.g., swaps between visually identical objects (Fig. 10). Furthermore, our method relies on pose estimation which can fail in some circumstances, such as extreme low-light, textureless surfaces, or sparse views with low co-visibility. Lastly, we focus on object-level spatial changes; semantic state changes (e.g., bread to toast) or fine-grained surface deformations (e.g., cracks) remain future work.

Acknowledgement This work is supported in part by NSF IIS grant 2312102. S.W. is supported by NSF 2331878 and 2340254, and research grants from Intel, Amazon, and IBM. This research used both the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois, and the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta and DeltaAI are joint efforts of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications. Special thanks to Prachi Garg and Yunze Man for helpful discussion during project development, Bowei Chen, Zhen Zhu, Ansel Blume, Chang Liu, and Hongchi Xia for general advice and feedback on the paper, and Haoqing Wang and Vladimir Yesayan for data collection and annotation.

References

1. Alcantarilla, P.F., Stent, S., Ros, G., Arroyo, R., Gherardi, R.: Street-view change detection with deconvolutional networks. Autonomous Robots **42**(7), 1301–1322 (2018)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W.,

Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
3. Bore, N., Ekekrantz, J., Jensfelt, P., Folkesson, J.: Detection and tracking of general movable objects in large three-dimensional maps. IEEE Transactions on Robotics (2019)
4. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Surís, D., Ryali, C.K., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Radle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Doll'ar, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: SAM 3: Segment anything with concepts. In: International Conference on Learning Representations (ICLR) (2026)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9630–9640 (2021)
6. Chen, S., Yang, K., Stiefelhagen, R.: Dr-tanet: Dynamic receptive temporal attention network for street scene change detection. In: IEEE Intelligent Vehicles Symposium (IV). pp. 502–509. IEEE (2021)
7. Cho, K., Kim, D.Y., Kim, E.: Zero-shot scene change detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 39, pp. 2509–2517 (2025)
8. Fehr, M., Furrer, F., Dryanovski, I., Sturm, J., Gilitschenski, I., Siegwart, R., Cadena, C.: Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 5237–5244. IEEE (2017)
9. Fu, J., Du, Y., Singh, K., Tenenbaum, J.B., Leonard, J.J.: Robust change detection based on neural descriptor fields. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2817–2824. IEEE (2022)
10. Galappaththige, C.J., Lai, J., Windrim, L., Dansereau, D.G., Suenderhauf, N., Miller, D.: Multi-view pose-agnostic change localization with zero labels. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11600–11610 (2025)
11. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., Gan, C., de Melo, C., Tenenbaum, J.B., Torralba, A., Shkurti, F., Paull, L.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. IEEE International Conference on Robotics and Automation (ICRA) pp. 5021–5028 (2023)
12. Huang, R., Jiang, B., Zhao, Q., Wang, W., Zhang, Y., Guo, Q.: C-nerf: Representing scene changes as directional consistency difference-based nerf. arXiv preprint arXiv:2312.02751 (2023)
13. Jiang, B., Huang, R., Zhao, Q., Zhang, Y.: Gaussian difference: Find any change instance in 3d scenes. In: ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
14. Kapur, J.N., Sahoo, P.K., Wong, A.K.: A new method for gray-level picture thresholding using the entropy of the histogram. Computer vision, graphics, and image processing **29**(3), 273–285 (1985)
15. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: Mapanything: Universal

feed-forward metric 3d reconstruction. In: 2026 International Conference on 3D Vision (3DV). pp. 499–509. IEEE (2026)
16. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (TOG) **42**, 1 – 14 (2023)
17. Kim, J.W., Kim, U.H.: Towards generalizable scene change detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24463–24473 (June 2025)
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.B.: Segment anything. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) pp. 3992–4003 (2023)
19. Kuhn, A., Sormann, C., Rossi, M., Erdler, O., Fraundorfer, F.: Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In: 2020 International Conference on 3D Vision (3DV). pp. 404–413. IEEE (2020)
20. Langer, E., Patten, T., Vincze, M.: Robust and efficient object change detection by combining global semantic information and local geometric verification. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8453–8460. IEEE (2020)
21. Lee, J.Y., DeGol, J., Zou, C., Hoiem, D.: Patchmatch-rl: Deep mvs with pixelwise depth, normal, and visibility. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6158–6167 (2021)
22. Lin, C.J., Garg, S., Chin, T.J., Dayoub, F.: Robust scene change detection using visual foundation models and cross-attention mechanisms. In: International Conference on Robotics and Automation (ICRA). pp. 8337–8343. IEEE (2025)
23. Liu, M., Shi, Q., Marinoni, A., He, D., Liu, X., Zhang, L.: Super-resolution-based change detection network with stacked attention module for images with different resolutions. IEEE Transactions on Geoscience and Remote Sensing pp. 1–18 (2021)
24. Looper, S., Rodriguez-Puigvert, J., Siegwart, R., Cadena, C., Schmid, L.: 3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 8179–8186. IEEE (2023)
25. Lu, Z., Ye, J., Leonard, J.: 3dgs-cd: 3d gaussian splatting-based change detection for physical object rearrangement. IEEE Robotics and Automation Letters (RAL) **10**, 2662–2669 (2024)
26. Ma, X., Gong, Y., Wang, Q., Huang, J., Chen, L., Yu, F.: Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5732–5740 (2021)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. Springer, Cham (2020)
28. Nguyen, H.H., Rahmanzadehgervi, P., Mai, L., Nguyen, A.T.: Improving zero-shot object-level change detection by incorporating visual correspondence. In: Proceedings of the IEEE/CVF Conference on Winter Conference on Applications of Computer Vision (WACV) (2025)
29. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y.B., Li, S.W., Misra, I., Rabbat, M.G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski,

P.: DINOv2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 **abs/2304.07193** (2023)
30. Park, J.M., Jang, J.H., Yoo, S.M., Lee, S.K., Kim, U.H., Kim, J.H.: Changesim: Towards end-to-end online scene change detection in industrial indoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8578–8585. IEEE (2021)
31. Park, J.M., Kim, U.H., Lee, S.H., Kim, J.H.: Dual task learning by leveraging both dense correspondence and mis-correspondence for robust change detection with imperfect matches. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13749–13759 (2022)
32. Perrett, T., Darkhalil, A., Sinha, S., Emara, O., Pollard, S., Parida, K., Liu, K., Gatti, P., Bansal, S., Flanagan, K., Chalk, J., Zhu, Z., Guerrier, R., Abdelazim, F., Zhu, B., Moltisanti, D., Wray, M., Doughty, H., Damen, D.: Hd-epic: A highly-detailed egocentric video dataset. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 23901–23913 (2025)
33. Pomerleau, F., Krüsi, P., Colas, F., Furgale, P.T., Siegwart, R.Y.: Long-term 3d map maintenance in dynamic environments. IEEE International Conference on Robotics and Automation (ICRA) pp. 3712–3719 (2014)
34. Qian, J., Chatrath, V., Yang, J., Servos, J., Schoellig, A., Waslander, S.: Pocd: Probabilistic object-level change detection and volumetric mapping in semi-static scenes. In: Robotics: Science and Systems (RSS) (2022)
35. Ramkumar, V.R.T., Bhat, P., Arani, E., Zonooz, B.: Self-supervised pretraining for scene change detection. In: Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia. pp. 6–14 (2021)
36. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C.K., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R.B., Doll'ar, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. In: International Conference on Learning Representations (ICLR) (2025)
37. Sachdeva, R., Zisserman, A.: The change you want to see. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3993–4002 (2023)
38. Sachdeva, R., Zisserman, A.: The change you want to see (now in 3d). IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) pp. 2052–2061 (2023)
39. Sakurada, K., Okatani, T., Deguchi, K.: Detecting changes in 3d structure of a scene from multi-view images captured by a vehicle-mounted camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 137–144 (2013)
40. Sakurada, K., Shibuya, M., Wang, W.: Weakly supervised silhouette-based semantic scene change detection. In: IEEE International conference on robotics and automation (ICRA). pp. 6861–6867. IEEE (2020)
41. Schmid, L., Abate, M., Chang, Y., Carlone, L.: Khronos: A unified approach for spatio-temporal metric-semantic slam in dynamic environments. In: Proc. of Robotics: Science and Systems (RSS). Delft, Netherlands (July 2024)
42. Schmid, L., Delmerico, J., Schönberger, J.L., Nieto, J., Pollefeys, M., Siegwart, R., Cadena, C.: Panoptic multi-tsdfs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 8018–8024 (2022)

43. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
44. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. arXiv preprint arXiv:2508.10104 (2025)
45. Tang, Z., Fan, Y., Wang, D., Xu, H., Ranjan, R., Schwing, A.G., Yan, Z.: Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
46. Varghese, A., Gubbi, J., Ramaswamy, A., Balamuralidhar, P.: Changenet: A deep learning architecture for visual change detection. In: Proceedings of the European conference on computer vision (ECCV) workshops (2018)
47. Varghese, S., Gao, J., Hoskere, V.: Viewdelta: Scaling scene change detection through text-conditioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2797–2807 (2025)
48. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Niessner, M.: Rio: 3d object instance re-localization in changing indoor environments. In: Proceedings IEEE International Conference on Computer Vision (ICCV) (2019)
49. Wang, G.H., Gao, B.B., Wang, C.: How to reduce change detection to semantic segmentation. Pattern Recognition **138**, 109384 (2023)
50. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 2025 International Conference on 3D Vision (3DV). pp. 78–89. IEEE (2025)
51. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
52. Wang*, Q., Zhang*, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
53. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
54. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Scalable permutation-equivariant visual geometry learning. In: International Conference on Learning Representations (ICLR) (2026)
55. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21924–21935 (2025)
56. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
57. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5525–5534 (2019)
58. Yugay, V., Kersten, T., Carlone, L., Gevers, T., Oswald, M.R., Schmid, L.: Gaussian mapping for evolving scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18903–18912 (2026)

59. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. In: International Conference on Learning Representations (ICLR) (2025)
60. Zhu, L., Huang, S., Schindler, K., Armeni, I.: Living scenes: Multi-object relocalization and reconstruction in changing 3d environments. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2024)