Abstract

Daily indoor scenes often involve constant changes due to human activities. To recognize scene changes, existing change captioning methods focus on describing changes from two images of a scene. However, to accurately perceive and appropriately evaluate physical changes and then identify the geometry of changed objects, recognizing and localizing changes in 3D space is crucial. Therefore, we propose a task to explicitly localize changes in 3D bounding boxes from two point clouds and describe detailed scene changes, including change types, object attributes, and spatial locations. Moreover, we create a simulated dataset with various scenes, allowing generating data without labor costs. We further propose a framework that allows different 3D object detectors to be incorporated in the change detection process, after which captions are generated based on the correlations of different change regions. The proposed framework achieves promising results in both change detection and captioning. Furthermore, we also evaluated on data collected from real scenes. The experiments show that pretraining on the proposed dataset increases the change detection accuracy by +12.8% (mAP0.25) when applied to real-world data. We believe that our proposed dataset and discussion could provide both a new benchmark and insights for future studies in scene change understanding.

1. Introduction

Physical-world often involves continuous and numerous changes. For example, indoor scenes often experience quantity, location, and arrangement changes involving household appliances. Accurate perception and recognition of changing information is an essential capability for future AI systems to provide appropriate assistance to human users, such as providing up-to-date information of household appliances for people with physical disabilities. Scene change detection based on two observations of the same scene, either images or 3D scans, has previously been studied [1][2][3][4]. These studies focus on localizing changed parts from scenes, ignoring semantic context of change such as changed object type. Scene change captioning is an emerging research field aimed at generating language descriptions of changes. The majority of scene change captioning [5][6][7][8][9][10][11][12][13][14][15] utilize a pair of 2D images as inputs. However, most previous change captioning studies were conducted using datasets with limited visual complexity, such as CLEVR-Change [11] and CLEVR-Multi-Change [12]. These datasets consist of primitive shapes and solid color backgrounds. In addition, the precise physical scales and 3D shapes of changing regions are difficult to obtain from 2D images, even though understanding 3D context is essential in real world applications such as object grasping, navigation, and room rearrangements. Moreover, it is difficult to fully comprehend changes that are randomly spanned in 3D space from a 2D image observed from a single viewpoint.

To address the above-mentioned problems, we propose a novel task of scene change detection and captioning from dynamic 3D scans (Figure 1). Due to the uncertainty of human activities, it is often necessary to observe scenes from various viewpoints to identify scene changes. Therefore, we deal with two 3D scans observed from different routes and viewpoints, which we call dynamic 3D scans. In contrast to 2D images, observing various viewpoints allows capturing changes that randomly exist in a 3D space. More specifically, we conduct a fine-grained change understanding including the location, changed object type, change type, and the spatial relationships between objects and room from two registered 3D point clouds. To evaluate and measure progress, we build a synthetic dataset, Change Detection and Captioning from Dynamic Scans (DyS2Change). We use an existing 3D simulator AI2THOR [16], which consists of a series of simulated interior rooms that allow object interactions. DyS2Change contains a total of 120 scenes, 37,715 change pairs, and 661,345 captions, capable of diagnosing various capabilities in change understanding.

We also propose an approach to detect change regions
Figure 1. The illustration of the proposed task. We deal with the input of two dynamic scans, each of which is obtained after randomly routing a camera through the scene (see the observation route, the dotted lines indicate the observation route and the arrows indicate the observation direction). The obtained scans (point clouds) are used to detect observable changes between the two scans, as well as to provide a linguistic description for each change region to obtain a detailed understanding of the changes.

The microwave has been moved close to the faucet.
The fridge on the corner of the room has been opened.

The major contributions of this work are as follows. (1) We propose a new task and dataset, DyS2Change, aimed at detecting and describing multiple scene changes from dynamic 3D scans of indoor scenes. (2) We propose an end-to-end framework that simultaneously detects and describes changes. (3) We conduct a simulation-to-reality (sim2real) study in this task. The results of pretraining on the DyS2Change dataset show significant model performance improvements (+12.8% in mAP0.25), demonstrating the efficacy of DyS2Change in real-world applications.

2. Related Work

2.1. Change Understanding

Change detection is designed to recognize pixel-level changes in sequential scene views. Change detection methods using 2D images [1, 2, 3, 4, 17] and 3D environments [18, 19, 20, 21] have been widely discussed previously. Among them, Ku et al. [21] proposed Change3D, a dataset for change detection from 3D point cloud. However, those above-mentioned methods usually do not specify details of the changed contents, such as the change type (e.g., adding or disappearing). In this work, we address the task of scene change captioning to assess the ability to capture the detailed change content.

More recently, several studies focused on change captioning, which generates language descriptions of scene changes from 2D images [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] or 3D data [22, 23]. The authors of study [7] first introduced a small-scale dataset for describing changes that occurred in surveillance videos. The authors of studies [11] and [12] proposed single and multiple object change captioning and intended to recognize detailed scene changes, including attributes and spatial location of changed objects. However, they only discussed under a primitive scene setup using geometric shapes and solid color backgrounds. The authors of [22] and [23] also proposed change captioning from 3D point clouds of indoor scenes. However, they discussed observations from fixed camera positions and only considered single-object changes, limiting their efficacy for scenes containing occlusion or multiple object changes.

This study focuses on 3D contexts and performs change captioning by using two dynamic scene scans to mimic the human observations. Moreover, while existing methods have been evaluated on synthetic datasets with limited complexities [11, 12] or outdoor scenes [7], we propose a dataset made up of indoor environments with various scenes. Additionally, we explicitly localize changes while previous methods only discuss on attention maps [11, 12].
change in the scene. The desk lamp against the wall has been moved from its original position. The laptop on the bed has been closed. Someone removed the chair next to the bed.

• Detection only.
• Not capturing semantic context (e.g., changed object type).

2.2. 3D Scene Understanding

3D object detection is a crucial task that aimed at detecting objects from 3D scenes. ScanNet [24] and SUN RGB-D [25] are two widely used datasets, both consisting of various 3D scans of indoor scenes with object bounding box annotations. Qi et al. proposed a Hough transform voting-based method VoteNet [26] that is built on point cloud feature extractor PointNet++ [27], while H3DNet [28] introduced a set of geometric primitives for enhancing VoteNet. Recently, several studies have introduced transformers [29, 30, 31]. Pointformer [29] introduced local and global transformer modules to better deal with different scales. 3DETR [30] extended 2D transformer-based detector DETR [32] to allow processing of point clouds. The authors of [31] proposed a Voxel Set Transformer which regards point cloud processing as a set-to-set translation. Instead of focusing on object detection, we propose an end-to-end network that enables the use of different 3D object detectors for change detection and captioning.

Recently, various studies, such as embodied question answering [33, 34], vision language navigation [35], and referring expression comprehension [36, 37], have incorporated language into 3D environments. Similar to those mentioned above, we propose DyS2Change, which incorporates languages into 3D environments and facilitates describing and localizing scene changes. Additionally, several methods for generating language descriptions of 3D data have been proposed, including image captioning [38, 39], and dense captioning [40, 41]. In contrast to these tasks, which take a single scene as input, we focus on describing scene changes, which requires capturing the relationships between two scenes. Weihs et al. [42] introduced a task of visual room rearrangement in which an embodied agent restores changed objects to their initial positions, which also requires change recognition, but our method extends that process to include change captioning and localization.

3. Dataset

Indoor scenes often undergo constant changes brought about by consumption/replenishment of home expendables,
object movements, and room furniture arrangements. However, despite their significance, there have been few efforts aimed at understanding indoor scene change. Hence, this study proposes the first benchmark dataset for change captioning and detection within indoor scenes. Moreover, to generate a scale dataset at low cost, we build up our dataset upon a simulator AI2THOR, which includes 120 simulated rooms with a range of interactable objects, thereby resulting in a dataset with relatively high visual complexity.

3.1. Dataset Novelty

In Figure 3 and Table 1, we compare our dataset to several existing datasets and summarize their differences in terms of change detection [21] and captioning [7, 11, 12, 22]. The Change3D [21] change detection dataset is very limited in size and does not consider fine-grained object changes such as the change location. Spot-the-Diff [7], CLEVR-Change [11], CLEVR-Multi-Change [12], and Indoor Scene Change [22] describe detailed changes but either only discussed under 2D images [7, 11, 12] or fixed camera setup [22], with limited dataset size [7] or limited dataset complexity [11, 12], and neither of them explicitly localizes changed regions. In contrast, our dataset is the first attempt to facilitate fine-grained changes in indoor scenes that is capable of allowing both change descriptions and localization. Additionally, our dataset consists of various scenes and objects with relatively high dataset complexity. Moreover, we utilize dynamic 3D scans in which every scene observation is collected randomly from multiple camera views, which is similar to human observations.

3.2. Generation Process

**Dataset Setup.** Similar to existing change captioning datasets [11, 12], we consider the five critical atomic change types, including add, delete, move, open, and close objects. We introduce 21 changeable object categories defined in AI2THOR along with 19 object categories that are included in object relationships to describe various changes.

**Scene Change Generation.** The scenes are observed twice to generate change pairs in which one to four random changes were implemented between the two observations. To mimic human observations, for each observation, we generate a random route through the room for each circuit, during which the observing agent’s height is set to 1.8 meters and the agent conducts observations from the viewpoints of straight ahead, 30 degrees left/right, and 30 degrees up/down. We record the registered point cloud, object classes and positions, and bounding boxes of changed regions during each observation.

**Caption Generation.** Change captions are automatically generated from recorded change information, object positions, and 30 predefined change caption templates. In order to reflect object relationships and localization in captions, we consider two relationship types: object-room (including room corner, room center, and against the wall) to reflect object localization inside a room, and object-object relationships (including below, above, on, and near) to refer to a change object (target-object) using a nearby object (anchor-object). One caption template is shown below. All caption templates and the detailed relationship definitions are provided in the supplementary material.

The \(<\text{target-object}>\) that is \(<\text{relationship1}>\ <\text{anchor-object1}>\) was moved from its original location to \(<\text{relationship2}>\ <\text{anchor-object2}>\).

**Dataset Statistics.** We used 96 scenes for training and 24 scenes for the test. After removing instances without observable changes, we obtained 37,715 scene change pairs and 661,345 captions. As shown in Figure 3 and Table 1, our dataset is the first change captioning dataset to allow both change caption, detection, and includes multiple changes in one scene and object relationship descriptions in change captioning to identify the specific object instance within an object class.

4. Methodology

Existing change captioning methods mainly generate change captions from two images without explicit change localization. The change region is critical in change recognition and various downstream tasks, such as change object rearrangement [42]. Hence, we propose an end-to-end framework called dense caption change (DenseChangeCap) (Figure 3) that detects the change regions in 3D bounding boxes, predicts the changed object classes, and then generates a change caption for each change region. We deal with point clouds’ input observed before and after scene changes. DenseChangeCap details are provided in the following.

**Input Data.** The model input includes two registered point clouds with known camera positions observed before and after scene changes. Since the differences between point clouds are expected to be effective in determining change localization, we compute point cloud differences by extracting the changed points in the before and after change point clouds. More specifically, we extract points from the before and after change point clouds that have distances exceeding the before and after surface threshold (threshold is set to 0.05 meters). Then, we obtain “before” and “after” change point clouds and “before\after” and “before\before” point clouds. Next, we adjust the rate of the four different point clouds and down sample the adjusted point clouds to a total of N points. Each point is represented with four dimensions, including the 3D coordinates X, Y, and Z, and one dimension indicating the resource of the point (before, after, before\after, or after\before).

**Point Feature Extraction.** Unlike images, point clouds are difficult to be processed using conventional CNN structures due to their irregular forms. Here, in a manner similar
to VoteNet [26] and 3DETR [30], we use PointNet++ [27] as the backbone to obtain point features from point clouds. More specifically, given the input point cloud with \( N \) points, the PointNet++ hierarchically extracts point features from local to global regions, resulting in point features with \( M \times (3+C) \) dimensions, where the \( M \) is the downsampled point number, 3 is the 3D coordinates of each point, and \( C \) is the feature dimension of each point.

### 4.1. Region Feature Extraction

Inspired by Densecap [43], which generates captions from image region-based features, we introduce a region feature extractor to cluster point features to region features in order to perform change localization and captioning. In our experiments, two region feature extractors 3DETR [30] and VoteNet [26] are employed. For additional details about these models, please refer to [30] and [26], respectively.

3DETR [30] adopts a transformer encoder-decoder structure on the top of point features to perform object detection. The 3DETR model introduces a query structure in the transformer decoder, which is obtained by farthest point sampling and Fourier positional embedding. We adopted the 3DETR structure here and obtained \( K \times (3+C) \)-dimensional features, where \( K \) is the number of queries.

VoteNet [26] introduces a Hough transform voting mechanism for clustering region features on top of point features and obtains \( M \times (3+C) \) votes from the point features. Next, the VoteNet further applies the set aggregation operation introduced in PointNet++, resulting in \( K \) clusters with \( (3+C) \) dimensions.

### 4.2. Detection Head and Captioner

Given \( K \times (3+C) \)-dimensional region features, we use a detection head to perform change detection and a captioner to generate a change caption for each change region. More specifically, the detection head conducts 3D bounding box regression and predicts the change type of each region as well as the changed object class. Here, similar to 3DETR, we predict bounding box information \( \hat{b} \) as \( \hat{b} = [\hat{c}, \hat{d}, \hat{s}, \hat{o}] \), where \( \hat{c}, \hat{d} \in [0, 1]^3 \) represents the center and size of bounding boxes, \( \hat{s} = [0, 1]^{D_{\text{change}}} \) is the probability distribution over \( D_{\text{change}} \) change types, and \( \hat{o} = [0, 1]^{D_{\text{object}}} \) is the probability distribution over \( D_{\text{object}} \) object types.

We introduce two different captioners based on LSTM and Transformer. Captioners generate probability distributions over vocabulary for each word of the change caption sentence of each change region. At each step \( t \), the input is a region feature with \( 3+C \) dimensions and the hidden state \( h_{t-1} \). The LSTM-based captioner generates the words of each sentence step-by-step. We also adopt a standard transformer decoder for caption generation, in which a masked self-attention and a feed-forward network are used to process the sentence features.

### 4.3. Loss Function.

The DenseChangeCap simultaneously conducts change detection, change object recognition, and change captioning. For change detection, we adopt the bipartite set matching used in DETR [32] and 3DETR [30], as the following:

\[
\text{Loss}_{\text{det}} = \lambda_c \| \hat{c} - c \|_1 + \lambda_d \| \hat{d} - d \|_1 - \lambda_s \hat{s}^T \log \hat{s} \quad (1)
\]

We adopt standard cross-entropy loss \( \text{Loss}_{\text{obj}} \) for change object recognition and \( \text{Loss}_{\text{cap}} \) for change captioning. The final loss has three terms as the following:

\[
\text{Loss} = \lambda_1 \text{Loss}_{\text{det}} + \lambda_2 \text{Loss}_{\text{obj}} + \lambda_3 \text{Loss}_{\text{cap}} \quad (2)
\]

**Implementation Details.** We randomly sample a total of 20,000 points as inputs during all experiments. In the base
experiments, we set the rate of point cloud differences in
the total point cloud to 90% (before\after and after\before)
and 10\% for before and after change point clouds and set the
object query number to 128. For change caption, we select
query features during the training process using the region
closest to the center of the ground truth change region. For
testing, we use the detected change region for captioning.
We train each model for 100 epochs on the training and
evaluate via the testing set. For 3DETR implementation,
we set the encoder head layer to 3 and 4 and the decoder
to 8 and 4. We implement LSTM captioners with two lay-
ers and 512 hidden dimensions, and transformer captioners
with two layers and two heads, and 2048 feed-forward di-
mensions. The weights of different losses in Equations (1)
and (2) are set as: $\lambda_c = 1$, $\lambda_d = 1$, $\lambda_s = 0.1$, $\lambda_1 = 1$,
$\lambda_2 = 0.1$, and $\lambda_3 = 0.1$.

5. Experiment

5.1. Datasets.

We use the proposed DyS2Change dataset to evaluate
models’ performance in change localization and caption-
ing. DyS2Change is built based on the AI2THOR simu-
lator, and the scenes included are purely synthetic. Accord-
ingly, to evaluate the efficacy of the DyS2Change dataset
when applied to real-world environments, we created a sepa-
rate dataset named SUNRGBD2Change based on the 3D
scan dataset SUN RGB-D [25], which was collected using
RGB-D cameras in real-world indoor environments.

We chose four object classes: “bed”, “chair”, “desk”,
“sofa” from the SUN RGB-D dataset and implemented
“add”, “delete”, “move” changes for the four object classes.
We introduced the “close” relationship to describe the ob-
ject spatial relationships. For each scene pair, we randomly
generated one to three changes. After the above processes,
we obtained the SUNRGBD2Change dataset with 6,425
change pairs (3,326 for training and 3,099 for testing) and
84,565 captions. Two examples are shown in Figure 4.

5.2. Experimental Setup

Evaluation Metrics. The change detection performance
of each model was evaluated through 3D detection evalua-
tion metrics mAP0.25, mAP0.5, mAR0.25, and mAR0.5,
where mAPs determine intersections over union above 25\%
and 50\%. Similarly, mARs consider the average recall. For
object classification and change caption, we introduce the
m@kIoU proposed in [40]. The m in m@kIoU stands for
the accuracy for object classification and four different cap-
tioning evaluation metrics BLEU [44], CIDER [45], ME-
TEOR [46], and ROUGE [47], which evaluate the similari-
ties between ground truth and predicted captions.

Baselines. We consider the two baselines in compar-
sion with the proposed DenseChangeCap method. In detail,
we implemented Baseline3D by directly adding an LSTM
captioner to VoteNet to allow a caption to be generated for
each region proposal. The Baseline3D input was set to
50\% of before and 50\% of after change point clouds with-
out using the point cloud differences. We also introduced
three state-of-the-art 2D image-based methods DUDA [11],
MCCFormers-D and MCCFormers-S [12] during the cap-
tioning module evaluation, and set the before and after
change image to the scene top view to allow change cap-
tion generation from image pairs.

5.3. Experiments on DyS2Change Dataset

In this subsection, we conducted experiments on
DyS2Change to evaluate change captioning and localiza-
tion of different model designs.

Base Experiments. As shown in Table 2 compared
with Baseline3D, our DenseChangeCaps obtained better
performance for both change detection (mAPs, mARs) and
change caption (B, C, M, R@0.25IoU), indicating the ef-
ficacy of DenseChangeCaps in this task. Also, we found
that both 3DETR and VoteNet detectors obtained similar
performance levels, although 3DETR showed higher per-
formance when IoU=0.25, while the VoteNet-based mod-
els were better when IoU=0.5. For change captioning eval-

Figure 4. Two dataset examples chosen from the SUNRGBD2Change dataset.
We provide the features of ground truth bounding boxes to DenseChangeCaps and found that, compared with Baseline2Ds, our methods performed better in change captioning, thereby indicating the superiority of using 3D information in this task. Additionally, we found the transformer-based methods provided the same level of performance with the LSTM structure, which might be because the proposed dataset consists of language generated from templates and the caption complexity is limited.

Evaluation of Different Change Types. We provide the model performance for different change types in Table 2 (right five columns). Baseline2Ds obtained relatively the same performance levels for different change types. On the contrary, all DenseChangeCaps perform worse for add and delete changes when compared to open, close, and move changes. In our dataset setup, we detect two separated bounding boxes for open, close, and move changes but only detect one bounding box for add and delete changes, thus making the learning examples relatively less prominent in those two change types.

Rate of Point Cloud Difference. We experimented on models with different point cloud difference rates in Figure 5 (left). Here, we found that compared with models without point cloud differences (0%) or without original point clouds (100%), using a portion (50% and 90%) of the point cloud differences could improve the model performance, demonstrating the efficacy of point cloud differences in localizing changes.

Query number. We experimented on different query numbers using DenseChangeCap with 3DETR detector and Transformer captioner in Figure 5 (right). The models obtained higher performance with query numbers of 64 and 128 when compared with query numbers of 16 and 32, indicating that increasing the query number could improve the model performance. However, the model performance with query number of 256 is decreased, revealing that there could be an upper bound of performance when the query number keeps increase.

Qualitative Results. Figure 6 shows experimental results of DenseChangeCap with 3DETR detector and LSTM captioner. In these examples, the proposed method correctly detected most of the change regions and generated content-related change captions. However, we also noticed that there is still room for improvement, such as in the detection of small objects, where the plunger in example (2) was not detected, and in captioning accuracy regarding the change types (e.g., “missing” and “moved” in example (1))
and spatial relationships (e.g., “near the bed” and “near the toilet” in example (2)). We believe that the model’s performance could be strengthened by improving object detection.

5.4. Sim2Real on SUNRGBD2Change Dataset

To evaluate the efficacy of the DyS2Change when applied to real-world environments, we evaluated the sim2real performance of models pretrained on DyS2Change dataset and then adopted those models to SUNRGBD2Change dataset, which consists of 3D scans of actual houses.

We conducted experiments on DenseChangeCap with a 3DETR detector and compared models with/without pretraining on the DyS2Change dataset. DyS2Change dataset pretraining was performed for 60 epochs, after which models were trained on the SUNRGBD2Change dataset for 10 epochs. The experimental results are shown in Table 4. Here, we found that models pretrained on the DyS2Change dataset outperformed models trained from scratch on SUNRGBD2Change dataset by large margins (with maximum +12.8% on mAP0.25 and +14.6% on BLEU). Even though the SUNRGBD2Change dataset consists of point cloud data with RGB-D camera sensor noise, the experimental results indicate the efficacy and potential of our proposed DyS2Change dataset and the applicability of our methods to real-world environments.

Table 4. Sim2Real study on SUNRGBD2Change dataset.

<table>
<thead>
<tr>
<th>Network Detector</th>
<th>Dataset Pre-training</th>
<th>Change Detection (mAP0.25)</th>
<th>Change Captioning (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DETR LSTM</td>
<td>DyS2Change</td>
<td>39.3</td>
<td>30.9</td>
</tr>
<tr>
<td>3DETR LSTM</td>
<td>DyS2Change</td>
<td>49.7</td>
<td>45.5</td>
</tr>
<tr>
<td>3DETR Transformer</td>
<td>DyS2Change</td>
<td>49.8</td>
<td>44.8</td>
</tr>
<tr>
<td>3DETR Transformer</td>
<td>DyS2Change</td>
<td>49.8</td>
<td>44.8</td>
</tr>
</tbody>
</table>

6. Conclusion

This paper proposed a novel task and a dataset for change detection and localization from dynamic 3D scans of indoor scenes. Existing change captioning methods do not explicitly detect change regions and were mainly evaluated on datasets with primitive objects and solid color background. Moreover, existing studies focus on 2D image pairs, limiting the model’s performance in room-scale change recognition. To resolve these issues, we proposed change captioning and localization from dynamic 3D scenes and a dataset with various indoor scenes. We also proposed an end-to-end framework that can incorporate various 3D object detectors and achieved promising results in the task. The experimental results also suggest the effectiveness of pretraining on proposed dataset for real-world application. We hope that our study can provide a benchmark in dynamic scene understanding and change recognition in 3D space.
References


