Zero-Shot Scene Change Detection

Kyusik Cho[1], **Dong Yeop Kim**[2,1], **Euntai Kim**[1*]

[1]Yonsei University, Seoul, Republic of Korea
[2]Korea Electronics Technology Institute, Seoul, Republic of Korea
ks.cho@yonsei.ac.kr, sword32@keti.re.kr, etkim@yonsei.ac.kr

Abstract

We present a novel, training-free approach to scene change detection. Our method leverages tracking models, which inherently perform change detection between consecutive frames of video by identifying common objects and detecting new or missing objects. Specifically, our method takes advantage of the change detection effect of the tracking model by inputting reference and query images instead of consecutive frames. Furthermore, we focus on the content gap and style gap between two input images in change detection, and address both issues by proposing adaptive content threshold and style bridging layers, respectively. Finally, we extend our approach to video, leveraging rich temporal information to enhance the performance of scene change detection. We compare our approach and baseline through various experiments. While existing train-based baseline tend to specialize only in the trained domain, our method shows consistent performance across various domains, proving the competitiveness of our approach.

Code — https://github.com/kyusik-cho/ZSSCD

Introduction

Scene Change Detection (SCD) is the task that aims to detect differences between two scenes separated by a temporal interval. Recently, SCD has gained significant interest in various applications involving mobile drones and robots. For instance, drone-based SCD has been studied for land terrain monitoring (Lv et al. 2023; Song et al. 2019; Agarwal, Kumar, and Singh 2019), construction progress monitoring (Han et al. 2021), and urban feature monitoring (Chen et al. 2016). Additionally, SCD using mobile robots has also been researched for natural disaster damage assessment (Sakurada and Okatani 2015; Sakurada, Okatani, and Deguchi 2013), urban landscape monitoring (Alcantarilla et al. 2018; Sakurada, Shibuya, and Wang 2020), and industrial warehouse management (Park et al. 2021, 2022).

Recently, SCD has been tackled using deep learning. Deep learning-based SCD techniques follow a procedure of learning from a training dataset and applying the model to a test dataset. These approaches tend to face two main challenges: dataset generation costs and susceptibility to style variations. Firstly, creating a training dataset for SCD models is labor-intensive and costly. Recent research has focused on reducing these costs through semi-supervised (Lee and Kim 2024; Sun et al. 2022) and self-supervised learning (Seo et al. 2023; Furukawa et al. 2020) methods, as well as the use of synthetic data (Sachdeva and Zisserman 2023; Lee and Kim 2024). While these approaches mitigate the expense of labeling, they often overlook the cost of acquiring image pairs, which arises from the substantial temporal intervals to capture changes. Secondly, due to the substantial temporal intervals between pre-change and post-change images, variations in seasons, weather, and time introduce significant differences in their visual characteristics. Consequently, SCD techniques must be robust to these style variations to be effective. However, the training dataset cannot include all the style variations present in real-world scenarios, making the trained model vulnerable to style variations that are not included in the training set.

To address these problems, we propose a novel training-free zero-shot SCD method. Our method does not require a training dataset, thereby eliminating collection costs and allows it to be applied to any problem with arbitrary styles. To the best of our knowledge, this paper is the first to attempt zero-shot SCD without training on SCD datasets. The key idea of this paper is to formulate SCD as a tracking problem, and apply a foundation tracking model to conduct zero-shot SCD. This idea stems from the observation that the tracking task is fundamentally similar to change detection. Specifically, tracking models (Cheng et al. 2023a; Cheng and Schwing 2022) maintain or build tracks by identifying the same objects, disappeared objects, and newly appeared objects in two consecutive images, even when the camera and objects move. Thus, if the two consecutive images in tracking are replaced with the two images before and after the change in SCD, the tracking model can automatically solve SCD without training.

However, there are some differences between tracking and SCD tasks: (a) Unlike tracking, the two images before and after the change in SCD might have different styles due to a large time gap between two images. We refer to this SCD trait as the style gap. (b) Objects change very little between two consecutive images in tracking, whereas objects change abruptly in SCD. We refer to this SCD trait as the content gap. To address these issues in our zero-shot SCD
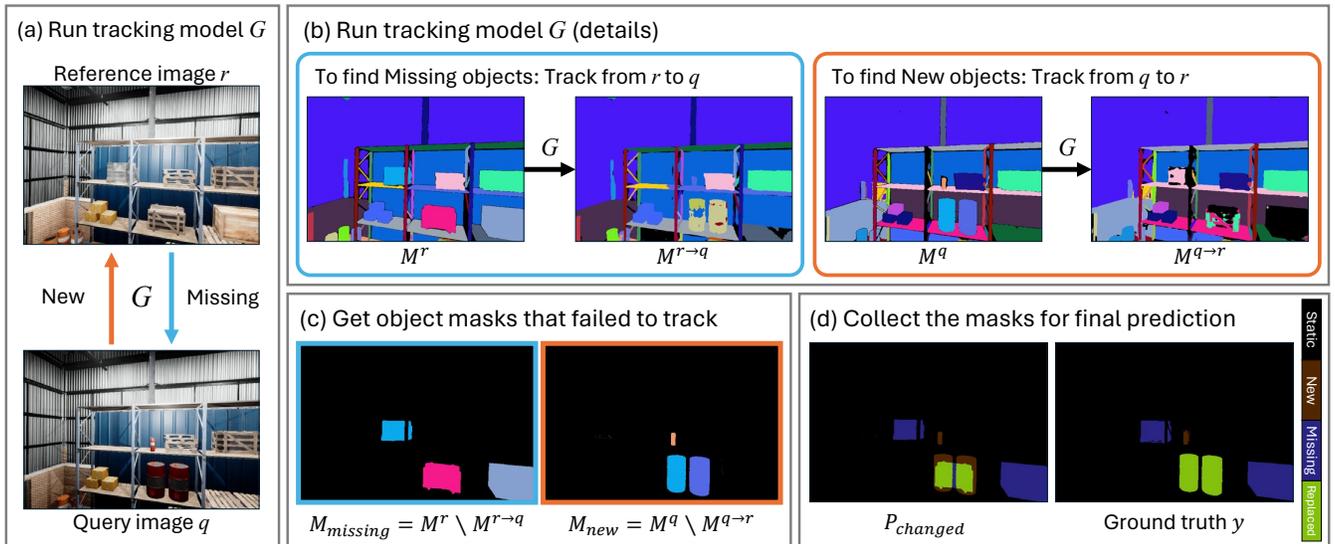
Figure 1: **The basic idea of SCD with tracking model.** (a) We execute the tracking model G with r and q. (b) We denote the tracking result from r to q as $M^{r \to q} = G(r, q, M^r)$, and the tracking result from q to r as $M^{q \to r} = G(q, r, M^q)$. (c) 'Missing' objects are the objects that exist in r but not in q. Therefore, we compare M^r and $M^{r \to q}$ to find missing objects. Conversely, 'new' objects are identified by comparing M^q and $M^{q \to r}$. (d) The final prediction is the simple combination of new and missing.

method, we introduce a style bridging layer and a content threshold, respectively.

Finally, we extend our approach to sequence, to introduce the zero-shot SCD approach works on the video. Since our approach operates based on a tracking model, it can be seamlessly extended to work with video sequences. The proposed zero-shot SCD approach has been evaluated on three benchmark datasets and demonstrated comparable or even superior performance compared to previous state-of-the-art training-based SCD methods.

Related Work

Scene Change Detection (SCD) In recent years, numerous deep learning-based change detection methods have been proposed for SCD. DR-TANet (Chen, Yang, and Stiefelhagen 2021) utilized attention mechanism based on the encoder-decoder architecture. SimSaC (Park et al. 2022) developed a network with a warping module to correct distortions between images. C-3PO (Wang, Gao, and Wang 2023) developed a network that fuses temporal features to distinguish three change types. However, various studies have aimed to address the challenge of obtaining data. For instance, Lee and Kim (2024) and Sun et al. (2022) have introduced semi-supervised learning, while Seo et al. (2023) and Fukawa et al. (2020) proposed the self-supervised learning with unlabeled data. However, Sachdeva and Zisserman (2023) and Lee and Kim (2024) utilized synthetic data to effectively increase the dataset. Despite these methods effectively reducing the label costs, they tend to overlook the cost of collecting image pairs. Moreover, the robustness against style change has not been previously discussed. An effective SCD method should be able to focus on content

changes regardless of variations in image style. However, since a single dataset cannot encompass all possible style variations, the performance tends to be specialized for the styles present in the dataset. This issue, while not evident in controlled laboratory environments, becomes a significant problem in real-world applications. Therefore, we propose a novel SCD method that does not rely on datasets. Our method operates without a training dataset, thus ensuring independence from specific styles.

Segmenting and Tracking Anything Recently, Segment Anything (Kirillov et al. 2023) has demonstrated highly effective performance in universal image segmentation. SAM has shown the ability to perform various zero-shot tasks, and has served as the foundational model for various studies (Peng et al. 2023; Maquiling et al. 2024). Building upon this research, researchers have explored various methods to extend its application to tracking. For example, SAM-Track (Cheng et al. 2023b) implemented tracking by combining SAM with the DeAOT (Yang and Yang 2022) mask tracker. SAM-PT (Rajič et al. 2023) integrated SAM with point tracking to develop the pipeline. DEVA (Cheng et al. 2023a) proposed a pipeline that uses the XMem tracker (Cheng and Schwing 2022) to track provided masks without additional training. Among various studies, we adopted DEVA with SAM masks as our tracking model, to achieve track-anything for SCD without further training.

Method

Each datum for scene change detection (SCD) is represented as a triplet (r, q, y), where r and q denote paired images acquired at distinct times t_0 and t_1, respectively, and y represents the change label between the image pair. The primary

objective of this task is to discern the scene change between the images captured at t_0 and t_1 when inspecting the latter. Herein, we call the image obtained at t_0 (r) as the reference image and the image acquired at t_1 (q) as the query image.

To perform scene change detection without training, our methodology integrates two pretrained models: a segmentation model F and a tracking model G. The segmentation model F segments images in an unsupervised manner, while the tracking model G tracks each mask generated by F across multiple images. We employ the Segment Anything Model (SAM) (Kirillov et al. 2023) as the segmentation model F and DEVA (Cheng et al. 2023a) for the tracking model G. Comprehensive details about parameters for F and G, and details about the mask generation process are provided in the supplementary materials.

The rest of the Method section is structured as follows: First, we introduce the basic idea for performing SCD between two images using F and G. Next, we discuss the differences between the tracking task and the SCD task, and then introduce methods to overcome these differences. Finally, we extend our approach to the video level.

Scene Change Detection with Tracking Model

Our approach uses two pretrained models, a segmentation model F and a tracking model G. The segmentation model F partitions image I into object-level masks, forming the set $M = \{m_1, m_2, \cdots, m_n\}$. There exists no overlap between distinct masks, that is, $m_i \cap m_j = 0, \forall\, i \neq j$. The tracking model G takes consecutive frame images (I^0, I^1) and the object masks of the first frame $M^0 = F(I^0)$ as input, and yields $M^{0 \rightarrow 1}$ as output, that is, $M^{0 \rightarrow 1} = G(I^0, I^1, M^0)$. Here, $M^{0 \rightarrow 1}$ represents the set of masks tracked from $M^0 = F(I^0)$ to I^1. By checking if each mask that was present in M^0 is also present in $M^{0 \rightarrow 1}$, we can determine which object masks in I^0 still exist or have disappeared in I^1.

The key idea of our zero-shot SCD approach is to apply a reference image r and a query image q instead of consecutive frames (I^0, I^1) to the tracking model G. Although the tracking model traditionally expects consecutive frames (I^0, I^1) for input, we deviate from this convention by providing reference image r and query image q instead. To avoid potential confusion, we rewrite the input and the output of the tracking model as $M^{r \rightarrow q} = G(r, q, M^r)$. By comparing the masks between M^r and $M^{r \rightarrow q}$, we identify object masks that exist at time t_0 but have disappeared at time t_1, corresponding to the 'missing' class in the change detection task. Specifically,

$$M_{missing} = M^r \setminus M^{r \rightarrow q}. \tag{1}$$

Additionally, we run the tracking model G again by reversing the order reference image r and query image q and feed them to G to obtain $M^{q \rightarrow r} = G(q, r, M^q)$. Similarly, we predict the 'new' objects by $M_{new} = M^q \setminus M^{q \rightarrow r}$, which represent the objects that appear at time t_1 but were absent at time t_0. Our pixel-wise prediction is obtained by applying the union of masks within M_{new} and $M_{missing}$. Pixels experiencing both 'new' and 'missing' occurrences are considered 'replaced.' Formally, change prediction $P_{changed}$ is

Query Image q Reference Image r Ground truth y

The ratio of mask area

$\frac{|m_1^{q \rightarrow r}|}{|m_1^q|} = 0.0113$

$\frac{|m_2^{q \rightarrow r}|}{|m_2^q|} = 0.$ $\frac{|m_3^{q \rightarrow r}|}{|m_3^q|} = 0.$

M^q $M^{q \rightarrow r}$

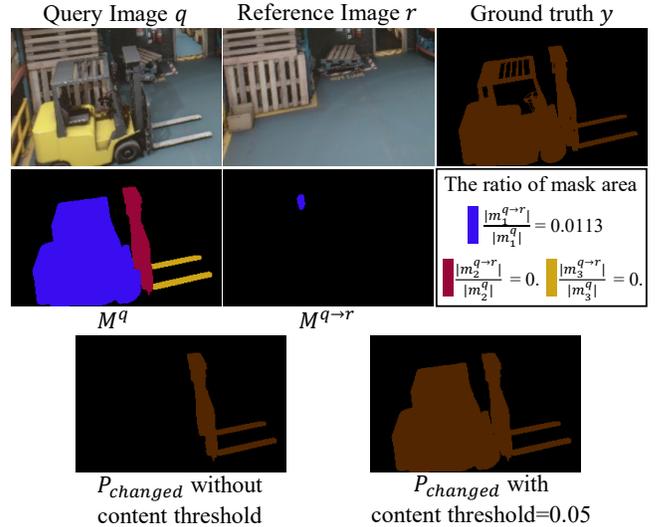$P_{changed}$ without content threshold $P_{changed}$ with content threshold=0.05

Figure 2: **Illustration of the content threshold.** Since the yellow forklift in q has disappeared in r, all the three masks (blue, red, and yellow masks) in M^q have no associated masks in $M^{q \rightarrow r}$. However, the tracking model creates a small area of the blue mask in $M^{q \rightarrow r}$ due to the content gap. This makes it mistakenly classified as a static object. To address this, we propose a content threshold to filter out masks whose area significantly reduces after tracking.

determined by:

$$
\begin{aligned}
P_{missing} &= \bigcup M_{missing} \\
P_{new} &= \bigcup M_{new} \\
P_{replaced} &= P_{missing} \cap P_{new} \\
P_{changed} &= P_{missing} \cup P_{new}
\end{aligned}
\tag{2}
$$

The entire process of this approach is illustrated in Figure 1.

Addressing Content Gap and Style Gap

As presented in the previous section, the key idea of our zero-shot SCD approach is to exploit the similarity between SCD and tracking tasks. However, directly applying this concept to various SCD scenarios leads to suboptimal performance due to inherent differences between the two tasks. In this section, we analyze these differences and propose corresponding solutions.

The first difference is the content gap, which refers to abrupt changes in content between the reference and query images. In traditional tracking tasks, objects typically disappear gradually over multiple frames rather than suddenly, and new objects appear gradually over multiple frames, implying that tracking tasks have little content gap. In contrast, SCD involves abrupt changes where objects disappear or appear within a single frame and it has a large content gap. Therefore, when the tracking model G trained on the tracking dataset is directly applied to SCD, the tracking model G tends to create small segments even for objects that have
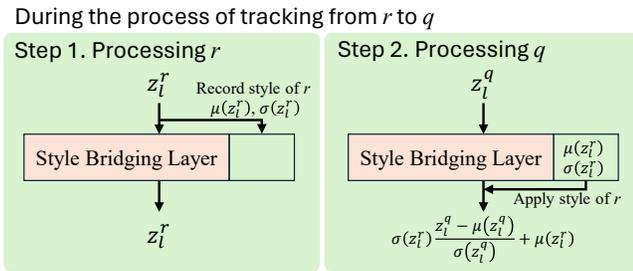
During the process of tracking from r to q

Step 1. Processing r	Step 2. Processing q

z_l^r Record style of r
$\mu(z_l^r), \sigma(z_l^r)$

Style Bridging Layer	

z_l^r

z_l^q

Style Bridging Layer	$\mu(z_l^r)$ $\sigma(z_l^r)$

Apply style of r

$\sigma(z_l^r)\dfrac{z_l^q - \mu(z_l^q)}{\sigma(z_l^q)} + \mu(z_l^r)$

Figure 3: **Illustration of the style bridging layer.** During the processing of the first image, the style is saved while the feature is passed through unchanged. When processing the second image, the saved style is applied to the feature.

disappeared, as shown in Figure 2. In the first row, the yellow forklift in the query image is missing from the reference image, but, in the second row, the mask in $M^{q \to r}$ tracked from a blue mask in M^q has a small segment. This remaining small segment makes the identification of missing objects very difficult.

To address the content gap, we propose considering an object as disappeared if its size is significantly reduced after tracking, even if it has not completely vanished. To the end, we introduce a content threshold τ and we compare the areas of the masks before and after tracking. If the ratio is less than the content threshold τ, we consider the corresponding object is missing or newly appeared. We define the $\setminus^\tau$ operator to replace the $\setminus$ operator in equation 1 as follows, where $|m_i^*|$ denotes the area of the i-th mask in set *:

$$A \setminus^\tau B := \{m_i^A \mid \frac{|m_i^B|}{|m_i^A|} < \tau, \quad \forall m_i^A \in A\}. \quad (3)$$

By introducing this operator, we have $M_{missing} = M^r \setminus^\tau M^{r \to q}$. The value of τ is automatically determined based on the input data. Further details and discussions on the determination of τ are provided in the supplementary material.

The second difference between tracking tasks and SCD is the style gap, which refers to the difference in style between the reference and query images. Change detection data have large temporal gaps, therefore lighting, weather, or season can change. These changes are commonly modeled as style changes (Tang et al. 2023). We define such variations as the style gap, which is not considered in traditional tracking tasks and can thus significantly degrade SCD performance.

To address the style gap, we introduce a style bridging layer (SBL) by incorporating an Adaptive Instance Normalization (AdaIN) layer (Huang and Belongie 2017) into the residual blocks of ResNet backbone (He et al. 2016) of the tracking model G. The AdaIN layer, widely used to reduce style differences between images across various fields (Karras, Laine, and Aila 2019; Wu et al. 2019; Xu et al. 2023), references the first image and applies its style to the second image, thereby reducing the style differences between the two images. Inspired by this, SBL addresses the style gap between two inputs without learning, with two

training-free style parameters. For example, during the process $M^{r \to q} = G(r, q, M^r)$, the style bridging layer records the mean and variance of each layer in image r and applies these statistics when processing image q. Formally, the style bridging layer updates the feature of z_l^q as follows, where z_l^* denotes the l-th layer feature of image *.

$$\tilde{z}_l^q = \sigma(z_l^r)\frac{z_l^q - \mu(z_l^q)}{\sigma(z_l^q)} + \mu(z_l^r). \quad (4)$$

The operation of the proposed style bridging layer is illustrated in Figure 3.

Through these two methods, we effectively and simply address the content gap and style gap. Note that the two improvements are also applied in the process $M^{q \to r} = G(q, r, M^q)$ and $M_{new} = M^q \setminus^\tau M^{q \to r}$.

Extension to the Video Sequences

In this subsection, we extend our image-based SCD approach to video sequences. Leveraging video data enhances spatial understanding by capturing scenes from multiple angles. The tracking model in our pipeline allows the seamless extension of our image-based SCD approach to video data.

Consider a video SCD dataset consisting of sequences of reference, query, and change labels, denoted as $\{r^t, q^t, y^t\}_{t=1}^T$, where T represents the length of the video sequence, and t denotes the time index. Compared to the image SCD, the video SCD requires two modifications. The first modification is simply to feed two sequences $\{r^t, q^t\}_{t=1}^T$ instead of image pair $\{r, q\}$ as input to the tracking model. Specifically, we start tracking to detect missing objects in the video with

$$\begin{aligned} M^{r^1 \to r^2} &= G(r^1, r^2, M^{r^1}) \\ M^{r^1 \to r^2 \to q^2} &= G(r^2, q^2, M^{r^1 \to r^2}) , \end{aligned} \quad (5)$$

and continue tracking throughout the entire video with

$$\begin{aligned} M^{r^1 \twoheadrightarrow r^t} &= G(r^{t-1}, r^t, M^{r^1 \twoheadrightarrow r^{t-1}}) \\ M^{r^1 \twoheadrightarrow q^t} &= G(r^t, q^t, M^{r^1 \twoheadrightarrow r^t}) , \end{aligned} \quad (6)$$

where $M^{r^1 \twoheadrightarrow r^t} = M^{r^1 \to r^2 \to \cdots \to r^{t-1} \to r^t}$ and $M^{r^1 \twoheadrightarrow q^t} = M^{r^1 \to r^2 \to \cdots \to r^{t-1} \to r^t \to q^t}$, as shown in Figure 4. In mask formulation, M^X denotes the output from the segmentation model F, whereas $M^{X \to Y}$ or $M^{X \twoheadrightarrow Y}$ denotes the output from tracking model G. This architecture is similar to the structure of the Bayes filters or Markov process (Thrun 2002), in that all the information processed from 1 to $t - 1$ are included in $M^{r^1 \twoheadrightarrow r^{t-1}}$. Consequently, $M^{r^1 \twoheadrightarrow r^t}$ can be incrementally updated from $M^{r^1 \twoheadrightarrow r^{t-1}}$ and r^t, q^t, without reprocessing all previous images. During the incremental update from $M^{r^1 \twoheadrightarrow r^{t-1}}$ to $M^{r^1 \twoheadrightarrow r^t}$, all functions for tracking, including updating feature memory and identifying new objects, are activated. Conversely, during the update from $M^{r^1 \twoheadrightarrow r^t}$ to $M^{r^1 \twoheadrightarrow q^t}$, these functions are deactivated. This processing sequence is designed to detect 'missing' objects, whereas the opposite processing sequence with swapping r and q is employed to detect 'new' objects.
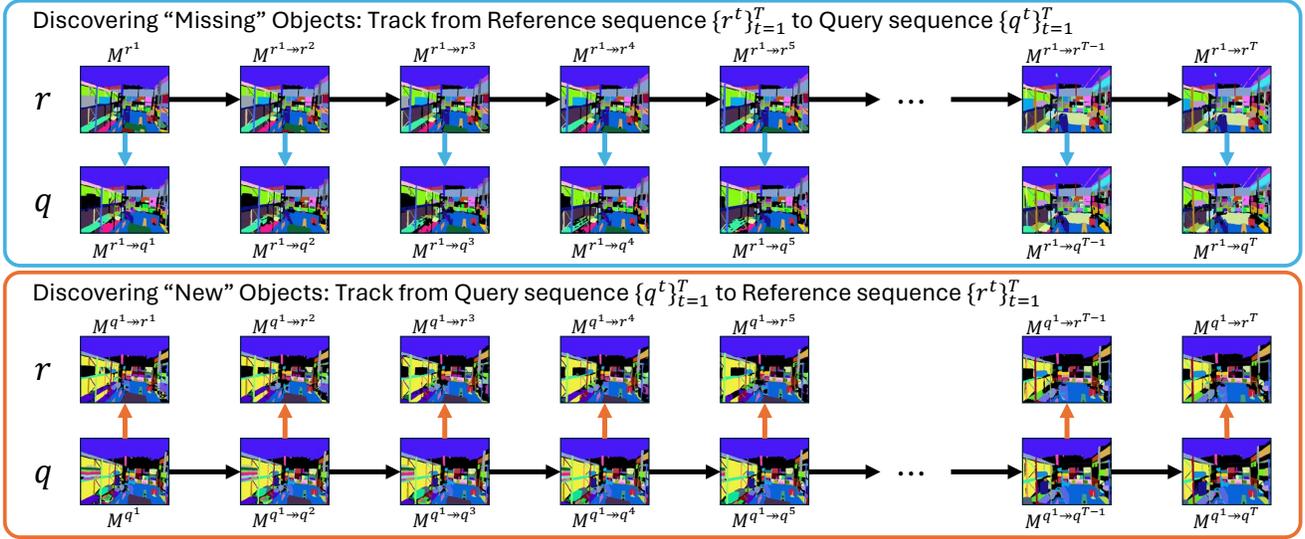
Figure 4: **Zero-shot SCD in video.** We conduct SCD on video sequences by providing sequence pairs instead of image pairs as input to the tracking model G. For each frame, the mask is propagated from the previous frame, resulting in a mask sequence through repeated propagation. SCD in the video is finalized by comparing the mask sequences.

The second modification is redefining $M_{missing}^t$ and M_{new}^t to suit video data. In equation 1, we defined $M_{missing}$ in image SCD as the masks that exist in the reference image r but are absent in the query image q. The definition of $M_{missing}^t$ for video SCD is an extension of this definition to video. Specifically, $M_{missing}^t$ is defined as the masks present in the reference sequence $\{r^t\}_{t=1}^T$ but absent in the query sequence $\{q^t\}_{t=1}^T$. Formally, $M_{missing}^t$ at time index t is defined by:

$$
\begin{aligned}
M_{missing}^t = \ & \{M^{r^t} \setminus^\tau M^{r^1 \twoheadrightarrow q^1}\} \\
& \cap \{M^{r^t} \setminus^\tau M^{r^1 \twoheadrightarrow q^2}\} \\
& \cap \ \cdots \\
& \cap \{M^{r^t} \setminus^\tau M^{r^1 \twoheadrightarrow q^T}\}.
\end{aligned}
\tag{7}
$$

M_{new}^t is similarly defined with the object that exists in query sequence $\{q^t\}_{t=1}^T$, but is absent in the reference sequence $\{r^t\}_{t=1}^T$.

Through these two extensions, our methodology becomes appropriate for processing videos. According to the definitions of these modifications, the sequence length can range from 1 to infinity. However, we impose an upper bound on the length of a sequence, denoted as T_{max}. If the length of the video exceeds T_{max}, it is divided into multiple sequences, each with a length of T_{max}. The reason for constraining the length of sequences is simple: as sequences lengthen, memory costs increase, while the relevant information for change detection decreases. For instance, in scenarios where the camera is in motion, the initial and final frames of a sequence may capture entirely different locations, rendering them unsuitable for change detection. Conversely, if the camera remains stationary, all frames depict the same scene, and additional frames provide redundant information. Therefore, we ensure more effective SCD by setting the upper bound of the sequence length. For our experiments, we set T_{max} to 60.

Experiments

Experimental Setup

In this section, we briefly introduce the datasets, the relevant settings, and the evaluation metrics.

ChangeSim (Park et al. 2021) is a synthetic dataset with an industrial indoor environment. It includes three subsets with varying environmental conditions: normal, low-illumination, and dusty air. The dataset categorizes changes into four types: new, missing, rotated, and replaced. Despite its variety of environmental variations and change classes, most baseline experiments on this dataset have evaluated only the binary change/unchange classification and have predominantly focused on the normal subset, leaving the dataset's full potential underexplored. Therefore, we chose the state-of-the-art method, C-3PO (Wang, Gao, and Wang 2023), and reproduced the results under the following conditions: using the original image size (640×480) to fully utilize the rich information; and including all three subsets. Among the four change classes in this dataset, the rotated class, unlike others, involves slight angular changes of the same object rather than complete appearances or disappearances. We considered this as the object remaining static and integrated this class into static for evaluation.

VL-CMU-CD (Alcantarilla et al. 2018) is a dataset that includes information on urban street view changes over a long period, encompassing seasonal variations. Following the baseline approach, we performed predictions using 512×512-sized images. As the change class in this dataset is limited to a binary classification of the 'missing' class, we used only the 'missing' class for our three types of prediction.

ChangeSim: In-domain							
Method	Trained Set	Test Set	Static	New	Missing	Replaced	mIoU
C-3PO	Normal	Normal	94.2	14.3	5.3	17.1	32.7
Ours	-		93.9	29.6	12.3	7.3	**35.8**
C-3PO	Dusty-air	Dusty-air	94.0	9.3	2.8	12.6	29.7
Ours	-		88.6	23.2	6.4	8.1	**31.6**
C-3PO	Low-illum.	Low-illum.	93.8	5.4	0.6	8.4	**27.1**
Ours	-		80.6	9.4	4.7	6.3	25.2

Table 1: **Experimental results on ChangeSim.** The results are expressed in per-class IoU and mIoU scores. Despite the absence of a training process, our model outperformed the baseline's in-domain performance in two out of three subsets.

ChangeSim: Cross-domain				
			Test set	
Method	Trained Set	Normal	Dusty-air	Low-illum.
C-3PO	Normal	32.7	27.2	26.7
	Dusty-air	29.6	29.7	26.9
	Low-illum.	29.4	27.1	**27.1**
Ours	-	**35.8**	**31.6**	25.2

Table 2: **Experimental results on ChangeSim, cross-domain.** The results are expressed in the mIoU score across all change classes. We trained the baseline model on each subset and tested it across all subsets. The experimental results show that the baseline model achieves the highest performance when the training set and test set are the same, while performance degrades when the training and test sets differ. In contrast, our method is free from this issue.

VL-CMU-CD & PCD				
			Test set	
Method	Trained Set	VL-CMU-CD	PCD	Average
C-3PO	VL-CMU-CD	**79.4**	11.6	45.5
	PCD	24.3	**82.4**	53.4
Ours	-	51.6	56.5	**54.0**

Table 3: **Experimental results on VL-CMU-CD and PCD.** The results are expressed in the F1 score. The baseline model performs best when the training and test are identical. However, its performance significantly declines when these datasets differ. Conversely, our method is robust to changes in the dataset, maintaining performance without the need for retraining whenever the test environment changes.

PCD (Sakurada and Okatani 2015) is a dataset consisting of panoramic images and includes two subsets: GSV and TSUNAMI. Following the baselines, we performed predictions on reshaped images of size 256×1024. Each data point is classified into binary change or unchanged categories. We consider the detected new, missing, and replaced predictions into a 'changed' class for evaluation.

Evaluation Metrics Following the previous work, we employ the mean Intersection over Union (mIoU) metric for ChangeSim and F1 score for VL-CMU-CD and PCD datasets.

Experimental Results

Table 1 presents the experimental results with other state-of-the-art, C-3PO (Wang, Gao, and Wang 2023) from ChangeSim. C-3PO is reproduced under the conditions described in the previous section. The baseline model is tested on the same datasets as its training dataset, denoted as in-domain. The table shows that our model achieved superior performance in two out of three subsets: normal and dusty-air, with mIoU of 35.8 and 31.6, respectively. However, our model shows lower performance in low-illumination subset, with mIoU of 25.2.

However, traditional train-based approaches are specialized for the style variation on which they were trained, becoming highly vulnerable when the domains differ. To illustrate this, we conducted additional experiments testing the baseline model on different domains from the ones it was trained on. Specifically, we tested the baseline model trained on a particular subset of ChangeSim on the other subsets, denoted as cross-domain. The experimental results are shown in Table 2. The baseline model, being specialized for their in-domain data, suffers performance drops when the data changes, indicating a lack of generalization. Specifically, when the baseline model trained on the dusty-air subset is tested on the same subset, it achieves a mIoU of 29.7. However, when the model is trained on the normal or low-illumination subsets and tested on the dusty-air subset, the mIoU scores are relatively lower, at 27.2 and 27.1, respectively. This performance drop is also observed in other cross-domain experiments. Conversely, our approach, which is not tailored to any specific style variation within the training dataset, shows robustness to changing the datasets. In other words, our model can be applied to all subsets without the need for retraining each time the environment changes.

We also conducted experiments on two additional real-world datasets, VL-CMU-CD and PCD. The results are summarized in Table 3. These results reveal a pattern consis-
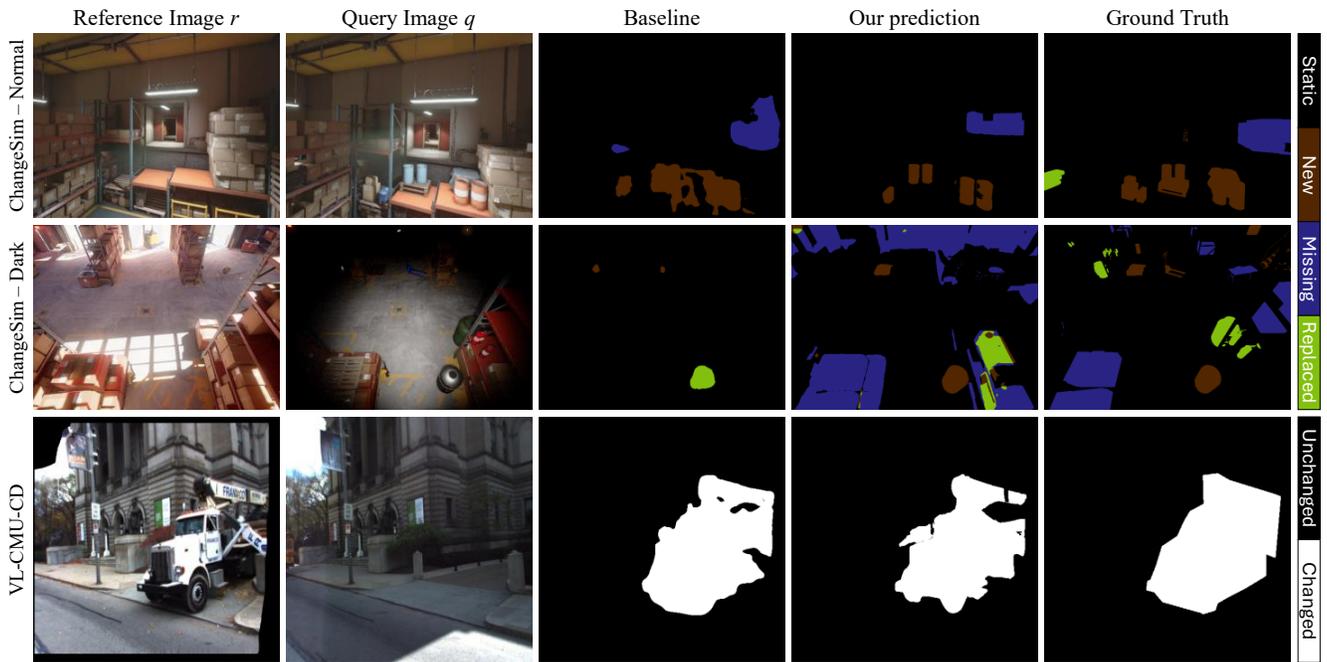
Figure 5: **Qualitative results.** Our approach successfully performs change detection across various datasets without training. For more qualitative results, see the supplementary material.

tent with our previous observations: the baseline model performs well when the training and test datasets are identical, but its performance significantly declines when the datasets differ. In this experiment, the performance drop is particularly severe due to the greater stylistic differences between the datasets. Specifically, the baseline model achieves high in-domain performance with F1 scores of 79.4 and 82.4 on the VL-CMU-CD and PCD datasets, respectively. In contrast, its cross-domain performance drops dramatically to 24.3 and 11.6, respectively. This observation suggests that the baseline model is overfitted to the limited style variations present in the training dataset.

Since the change detection method must function across various seasons and weather conditions in real-world scenarios, robustness to style changes is crucial for reliable performance. The most straightforward way to achieve such robustness against style variations is to employ an approach that does not rely on the training set, as supported by these experimental results.

We present qualitative results in Figure 5. The baseline results correspond to in-domain scenarios, where the training and test datasets are identical. In contrast, our predictions are zero-shot results without any training. The qualitative results show that our approach effectively performs change detection without training. In the supplementary material, we provide more qualitative results and intermediate outcomes of the prediction process.

Ablation Experiments

We conducted extensive ablation experiments and more analyses in the supplementary materials to validate the ef-

ficacy and robustness of our proposed methods. (1) Performance of zero-shot SCD with and without the proposed adaptive content threshold (ACT) and style bridging layer (SBL). (2) Ablation study based on different SBL settings. (3) Ablation study based on different clip lengths. (4) Ablation study based on different ACT settings.

Conclusion

In this paper, we present a novel approach to zero-shot Scene Change Detection (SCD) applicable to both image and video. Our method performs SCD without training by leveraging a tracking model, which inherently performs change detection between consecutive video frames by recognizing common objects and identifying new or missing objects. To adapt the tracking model specifically for SCD, we propose two training-free components: the style bridging layer and the adaptive content threshold. Through extensive experiments on three SCD datasets, our approach demonstrates its versatility by showing its robustness to various environmental changes. We believe that our work offers a fresh perspective on SCD and represents a significant step forward in its practical application.

Acknowledgments

References

Agarwal, A.; Kumar, S.; and Singh, D. 2019. Development of neural network based adaptive change detection technique for land terrain monitoring with satellite and drone images. *Defence Science Journal*, 69(5): 474.

Alcantarilla, P. F.; Stent, S.; Ros, G.; Arroyo, R.; and Gherardi, R. 2018. Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42: 1301–1322.

Chen, B.; Chen, Z.; Deng, L.; Duan, Y.; and Zhou, J. 2016. Building change detection with RGB-D map generated from UAV images. *Neurocomputing*, 208: 350–364.

Chen, S.; Yang, K.; and Stiefelhagen, R. 2021. Dr-tanet: Dynamic receptive temporal attention network for street scene change detection. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, 502–509. IEEE.

Cheng, H. K.; Oh, S. W.; Price, B.; Schwing, A.; and Lee, J.-Y. 2023a. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1316–1326.

Cheng, H. K.; and Schwing, A. G. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision (ECCV)*, 640–658. Springer.

Cheng, Y.; Li, L.; Xu, Y.; Li, X.; Yang, Z.; Wang, W.; and Yang, Y. 2023b. Segment and track anything. *arXiv preprint arXiv:2305.06558*.

Furukawa, Y.; Suzuki, K.; Hamaguchi, R.; Onishi, M.; and Sakurada, K. 2020. Self-supervised Simultaneous Alignment and Change Detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6025–6031.

Han, D.; Lee, S. B.; Song, M.; and Cho, J. S. 2021. Change detection in unmanned aerial vehicle images for progress monitoring of road construction. *Buildings*, 11(4): 150.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1501–1510.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4401–4410.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4015–4026.

Lee, S.; and Kim, J.-H. 2024. Semi-Supervised Scene Change Detection by Distillation From Feature-Metric Alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1226–1235.

Lv, Z.; Huang, H.; Sun, W.; Jia, M.; Benediktsson, J. A.; and Chen, F. 2023. Iterative training sample augmentation for enhancing land cover change detection performance with deep learning neural network. *IEEE Transactions on Neural Networks and Learning Systems*.

Maquiling, V.; Byrne, S. A.; Niehorster, D. C.; Nyström, M.; and Kasneci, E. 2024. Zero-shot segmentation of eye features using the segment anything model (sam). *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(2): 1–16.

Park, J.-M.; Jang, J.-H.; Yoo, S.-M.; Lee, S.-K.; Kim, U.-H.; and Kim, J.-H. 2021. Changesim: Towards end-to-end online scene change detection in industrial indoor environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8578–8585. IEEE.

Park, J.-M.; Kim, U.-H.; Lee, S.-H.; and Kim, J.-H. 2022. 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)*, 13749–13759.

Peng, Z.; Tian, Q.; Xu, J.; Jin, Y.; Lu, X.; Tan, X.; Xie, Y.; and Ma, L. 2023. Generalized Category Discovery in Semantic Segmentation. *arXiv preprint arXiv:2311.11525*.

Rajič, F.; Ke, L.; Tai, Y.-W.; Tang, C.-K.; Danelljan, M.; and Yu, F. 2023. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*.

Sachdeva, R.; and Zisserman, A. 2023. The Change You Want to See. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Sakurada, K.; and Okatani, T. 2015. Change detection from a street image pair using cnn features and superpixel segmentation. In *British Machine Vision Conference (BMVC)*.

Sakurada, K.; Okatani, T.; and Deguchi, K. 2013. Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 137–144.

Sakurada, K.; Shibuya, M.; and Wang, W. 2020. Weakly supervised silhouette-based semantic scene change detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6861–6867. IEEE.

Seo, M.; Lee, H.; Jeon, Y.; and Seo, J. 2023. Self-pair: Synthesizing changes from single source for object change detection in remote sensing imagery. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 6374–6383.

Song, F.; Dan, T.; Yu, R.; Yang, K.; Yang, Y.; Chen, W.; Gao, X.; and Ong, S.-H. 2019. Small UAV-based multi-temporal change detection for monitoring cultivated land cover changes in mountainous terrain. *Remote sensing letters*, 10(6): 573–582.

Sun, C.; Wu, J.; Chen, H.; and Du, C. 2022. SemiSANet: A semi-supervised high-resolution remote sensing image change detection model using Siamese networks with graph attention. *Remote Sensing*, 14(12): 2801.

Tang, G.; Ni, J.; Chen, Y.; Cao, W.; and Yang, S. X. 2023. An improved CycleGAN based model for low-light image enhancement. *IEEE Sensors Journal*.

Thrun, S. 2002. Probabilistic robotics. *Communications of the ACM*, 45(3): 52–57.

Wang, G.-H.; Gao, B.-B.; and Wang, C. 2023. How to reduce change detection to semantic segmentation. *Pattern Recognition*, 138: 109384.

Wu, Z.; Wang, X.; Gonzalez, J. E.; Goldstein, T.; and Davis, L. S. 2019. Ace: Adapting to changing environments for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2121–2130.

Xu, Q.; Zhang, R.; Wu, Y.-Y.; Zhang, Y.; Liu, N.; and Wang, Y. 2023. Simde: A simple domain expansion approach for single-source domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4797–4807.

Yang, Z.; and Yang, Y. 2022. Decoupling features in hierarchical propagation for video object segmentation. *Advances in Neural Information Processing Systems*, 35: 36324–36336.