STOW: Discrete-Frame Segmentation and Tracking of Unseen Objects for Warehouse Picking Robots

Anonymous Author(s)
Affiliation
Address
email

Abstract: Segmentation and tracking of unseen object instances in discrete frames pose a significant challenge in dynamic industrial robotic contexts, such as distribution warehouses. Here, robots must handle object rearrangements, including shifting, removal, and partial occlusion by new items, and track these items after substantial temporal gaps. The task is further complicated when robots encounter objects beyond their training sets, thereby requiring the ability to segment and track previously unseen items. Considering that continuous observation is often inaccessible in such settings, our task involves working with a discrete set of frames separated by indefinite periods, during which substantial changes to the scene may occur. This task also translates to domestic robotic applications, such as table rearrangement. To address these demanding challenges, we introduce new synthetic and real-world datasets that replicate these industrial and household scenarios. Furthermore, we propose a novel paradigm for joint segmentation and tracking in discrete frames, alongside a transformer module that facilitates efficient inter-frame communication. Our approach significantly outperforms recent methods in our experiments. For additional results and videos, please visit https://sites.google.com/view/stow-corl23.

Keywords: Unseen Object Instance Segmentation, Unsupervised Multi Object Tracking, Zero-shot, Discrete Frames

1 Introduction

Object segmentation and tracking is a key perception task for robotic picking. This is particularly important in warehouse environments, where millions of commodity items are organized daily on warehouse shelves for storage and categorization, as shown in Figure 1. Future intelligent robots must acquire strong perception capabilities to help human workers stow and fetch items from these shelves. Such capabilities include detecting objects with diverse geometry in cluttered scenes and tracking them while other items are added or picked up. While there has been surging industrial need supported by research progress on developing high-performance object instance segmentation and tracking systems, it is still notoriously difficult to detect and track a priori unknown objects in highly cluttered environments.

Figure 1: Depiction of a densely packed shelf environment. The shelf holds objects from a wide array of categories. During the stowing process, a human operator may obscure the camera's view and rearrange the objects within the bin. The robot's task is to pick a specific object as directed by the given order index of the object's placement in the bin.

Researchers commonly address this task using a *segment-then-track* paradigm [1, 2]. This paradigm involves executing the segmentation and tracking procedures sequentially. During segmentation, each frame is handled as an independent image, applying advanced unseen object instance segmen-

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

tation methods [3, 4, 5, 6]. The following tracking step aims to leveraging different techniques [7, 8, 9, 10] to group masks of the same object across frames. However, this approach has inherent limitations. In the segmentation phase, methods struggle to resolve ambiguities due to their inability to utilize information from other frames to enhance segmentation within each frame. As for the tracking phase, it lacks alternatives when the segmentation step fails, with the success of tracking heavily reliant on consistent object appearance or location across consecutive frames. These drawbacks limit the paradigm's effectiveness in situations involving crowded scenes and discrete frames.

Our task bears similarities with video instance segmentation (VIS), which involves segmenting and tracking objects in videos, with the leaderboard predominantly occupied by *simultaneous segmenting and tracking* methods [11, 12, 13]. This similarity suggests the potential of adopting their methods to realize an end-to-end solution in our task. However, these methods are mainly designed for videos with continuous frames. As a result, they display subpar performance when faced with significant object movements between frames, a prominent challenge in our task.

We introduce STOW, Discrete-Frame **S**egmentation and **T**racking of Unseen **O**bjects for **W**arehouse Picking Robots, a novel framework that addresses the said challenges. Employing a new paradigm to perform joint segmentation and tracking, and a novel module named multi-frame attention layer enabling efficient inter-frame communication, we succeed in achieving high segmentation accuracy, high tracking accuracy, and high robustness to the sim-to-real gap simultaneously. Remarkably, our method, even when trained exclusively on synthetic images, exhibits superior performance on real data and live robot experiments, marking considerable enhancements compared to its baseline counterparts.

In summary, our main contributions are: (1) We proposed a new paradigm to perform joint segmentation and tracking in discrete frames, along with a novel module named multi-frame attention that efficiently communicates information across frames; (2) We formulated the tasks of unseen object instance segmentation and tracking in discrete frames, generated realistic synthetic data, collected, and manually labeled a real dataset for bins in the shelf and tabletop environment, facilitating research in this domain; (3) We conducted experiments on real data and an actual robot, verifying our network's superior performance.

2 Related Works

Unseen object instance segmentation. In computer vision, traditional object instance segmentation requires prior knowledge about the objects. Our work, instead, targets potentially unseen objects without such assumptions. Previous efforts, such as UCN [4], UOIS [3], MF [5] have addressed unseen object discovery in single-frame images using varied strategies like RGB-D feature embeddings and metric learning loss. A recent model, SAM [6], also demonstrates robust segmentation across a wide variety of objects by training on a large amount of data. While these works still focus on unseen object discovery in single-frame images, our research expands this to segmenting and tracking unseen objects across multiple input frames.

Unsupervised multi-object tracking. Our proposed task shares similarities with the unsupervised video instance segmentation task introduced in DAVIS 2019 [14]. However, our task differs in two main aspects: (1) the need for perception systems to handle objects or categories that are not present in the training set, thus requiring the network to generalize to objects with varying shapes, whereas DAVIS 2019 still focuses on a large number of humans, vehicles, and animals; and (2) we deal with discrete frames, which poses a challenge for the network to accurately predict the location and mask of an object in the next frame with limited information from the previous frame.

Video object segmentation. We share the same propriety with video object segmentation task [15, 16, 9] that we all require to track a unseen object in the test set. However, the VOS task assumes the accessibility to the mask of the object in one frame, which is not applicable in our task. For video object instance segmentation, previous work following the *tracking by detection* paradigm, e.g. [17] and similarly [18, 19, 20], are often dealing with the problem of multi-object tracking (MOT), i.e. the estimation of bounding boxes and identities of objects in consecutive RGB image streams. MOT tasks usually focus on tasks in traffic scenes and deal with objects like persons [21, 22] and vehicles.

Image Co-segmentation. Our approach is similar to object co-segmentation [23], which extracts recurring objects from an image pair or a set of images. While the goal of co-segmentation is to identify shared objects in a scene the major difference to our approach is that we focus on instance segmentation and do not allow similar objects of the same class. Distinguishing between an object instance and an object class is a crucial requirement for industrial warehouses.

3 Problem Formulation

In this work, we introduce the novel task of segmenting and tracking unseen objects given a series of input discrete image frames. This task, while challenging, has broad applications in the field of robotics and is particularly useful in warehouse environments such as Figure 1. The scenes involving warehouse shelves can be exceedingly packed and cluttered, comprising a vast assortment of items, including some that may never have been encountered before. Moreover, there may be temporal gaps between successive snapshots of the scene, during which human workers may place new objects, robots may retrieve existing items, and some objects may undergo changes in pose.

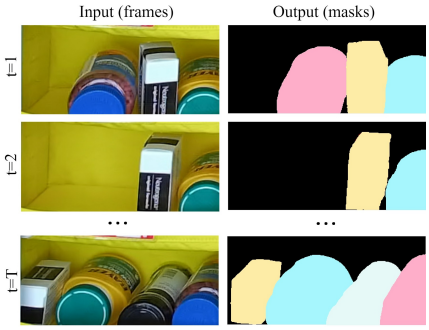

Figure 2: Illustration of our task.

Formally, we formulate the problem as follows. The input to the task is a sequence of images $\mathcal{I} = \{I_1, I_2, \cdots, I_T \mid I_t \in \mathbb{R}^{H \times W \times C_I}\}$ where H and W are the height and width of the images respectively and C_I represents the number of channels, which is three for RGB images and four for RGB-D inputs. The task involves detecting, segmenting, and tracking $K_\mathcal{I}$ object instances that appear in these input images, where $K_\mathcal{I}$ may not be known beforehand. The output of the task is a set of binary object instance masks, $\mathcal{M}_t = \{M_t^1, M_t^2, \cdots, M_t^{K_\mathcal{I}} \mid M_t^i \in \{0,1\}^{H \times W}, i = 1, 2, \cdots, K_\mathcal{I}\}$, corresponding to each input image $I_t \in \mathcal{I}$. Figure 2 provides a visual representation of the problem setting under consideration.

4 Method

Our system (Figure 3) utilizes query-based transformer architectures [24, 13] for object detection and segmentation, which we recapitulate in Sec. 4.1. To enable the tracking of object instances across discrete image frames, we introduce the learning of additional object embeddings for tracking (Sec. 4.2), as well as a multi-frame attention layer to distinguish between identical or distinct object instances (Sec. 4.3). Training specifics and loss functions are elaborated in Sec. 4.4.

4.1 Backbone Architecture

Following [24, 13], the backbone of our network primarily comprises three components: a ResNet-based image encoder, a transformer-based object query decoder, and several prediction heads tasked with determining object properties, such as the likelihood of object existence and object masks.

ResNet-based image encoder. We use ResNet-50 [25] as our backbone, which transforms every input frame $I_t \in \mathbb{R}^{H \times W \times C_I}$ $(t \in 1, 2, \ldots, T)$ into a dense low-resolution feature map $F_t \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times C_F}$. Here, C_F denotes the channel dimension of the output dense feature map, while $S = 32$ is the down-sampling ratio use in this work.

Transformer-based object query decoder. We employ a DETR-like transformer decoder [26] that takes the produced dense feature map F_t as input and learns to decode a set of N_q object tokens $\{\mathbf{q}_t^1, \mathbf{q}_t^2, \cdots, \mathbf{q}_t^{N_q}\} \in \mathbb{R}^{C_q}$ as the outputs. Each object token contains the latent information necessary for tasks such as classification estimation, mask prediction, and tracking in discrete frames. Our transformer decoder consists of $L = 10$ transformer blocks, each of which is composed of one cross-attention layer, one self-attention layer, one feed-forward layer, and one novel multi-frame attention layer that correlates object features across different image frames that we will describe in more detail in Sec. 4.3.

Prediction head for object masks. To get a per-pixel segmentation mask $M_t^i \in \{0,1\}^{H\times W}$ for each object token $\mathbf{q}_t^i$, we first use a two-layer multilayer perceptron (MLP) that maps an input object token q_t^i to a mask embedding $\mathbf{e}_t^i \in \mathbb{R}^{C_e}$, and then employ a multi-scale deformable attention transformer module (MSDeformAttn) [27] to convert the dense feature map F_t to a pixel embedding map $P_t \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C_e}$. Here, C_e denote the channel dimensions used for the mask and pixel embeddings. We calculate the dot product between the object embedding $\mathbf{e}_t^i$ and the pixel embedding P_t in order to obtain the mask prediction $\hat{M}_t^i$ at a reduced resolution of $\frac{H}{4} \times \frac{W}{4}$. Subsequently, a bilinear upsampling operator is applied to map $\hat{M}_t^i$ back to the original image resolution for the final mask prediction $M_t^i \in \{0,1\}^{H\times W}$.

Prediction head for object existence scores. Taking the object query $\mathbf{q}_t^i$ as input, we leverage a simple linear layer to estimate an object existence likelihood score $s_t^i \in [0,1]$. In the context of unseen object instance segmentation, we are not concerned with the precise categories of the target objects, nor do we possess this knowledge. This characteristic simplifies the classification task into a binary classification, the objective of which is to estimate the confidence score of whether each segment corresponds appropriately to an object.

4.2 Object Embedding for Tracking

In addition to the predicted object existence score and its segmentation mask in each frame, we add a new prediction head for object tracking embedding to enable the association of object tokens belonging to the same object from different input frames. More specifically, we employ a two-layer MLP to learn a mapping from the input object query $\mathbf{q}_t^i$ to another object embedding used for tracking $\mathbf{r}_t^i \in \mathbb{R}^{C_r}$. Here, C_r represents the number of channels of the object embedding vector.

During the inference phase, we implement an associator to group object tokens with similar object embeddings. Specifically, we sequentially traverse each frame in the sequence. For each frame, we retain only those object tokens $\mathbf{q}^i$ whose confidence scores surpass a predefined threshold δ_{score}. For each input sequence, we maintain a trajectory bank, $\mathcal{T}$, which encompasses the trajectories of all observed objects. Each trajectory in the trajectory bank, $\mathcal{T}^i \in \mathcal{T}$, comprises object tokens $\mathbf{q}^i$ that are considered as belonging to the object i. Subsequently, we compute the similarity score between these selected object tokens and the previous trajectory using the following equation:

$$Sim(\mathbf{q}^i, \mathcal{T}^j) = \max(R(\mathbf{q}^i) \cdot R(\mathbf{q}_k^j)), \text{for } \mathbf{q}_k^j \in \mathcal{T}^j \tag{1}$$

In this equation, R is the function that transforms the object token q^i into the object embedding r^i.

After this step, we utilize the Hungarian algorithm [28] to search for an optimal bipartite matching $\hat{\varrho}$ from all possible bipartite matching $\mathcal{P}$ that can maximize the overall similarity between object tokens $\mathbf{q}$ and the trajectory bank $\mathcal{T}$:

$$\hat{\varrho} = \arg\max_{\varrho\in\mathcal{P}} \sum_{i}^{N_q} Sim(\mathbf{q}^i, \mathcal{T}^{\varrho(i)}) \tag{2}$$

We initialize the trajectory bank, $\mathcal{T}$, with an adequate number of false alarm tokens, $\mathcal{T}_{FA}$, each of which exhibits a constant similarity δ_{match} to all object tokens. The hyper-parameter δ_{match} can also be interpreted as a false alarm threshold in matching. Any predicted object tokens assigned to tokens from $\mathcal{T}_{FA}$ are regarded as not matching with any existing trajectory and thus, a new trajectory is opened for them.

Our tracking method also enables the possibility to handle multiple identical objects in the same scene. Specifically, object tokens corresponding to identical objects are likely to be recognized due to the expectation of object embeddings close in distance for such object tokens.

4.3 Multi-Frame Attention

The previous modules we introduced work independently for each frame without any cross-frame information exchange. To bridge this gap and facilitate efficient communication between frames, we introduce a novel component, the multi-frame attention layer, into the transformer decoder.

The multi-frame attention layer is an extension of the self-attention layer. While the self-attention layer in our transformer decoder operates on object queries from a single frame, the multi-frame

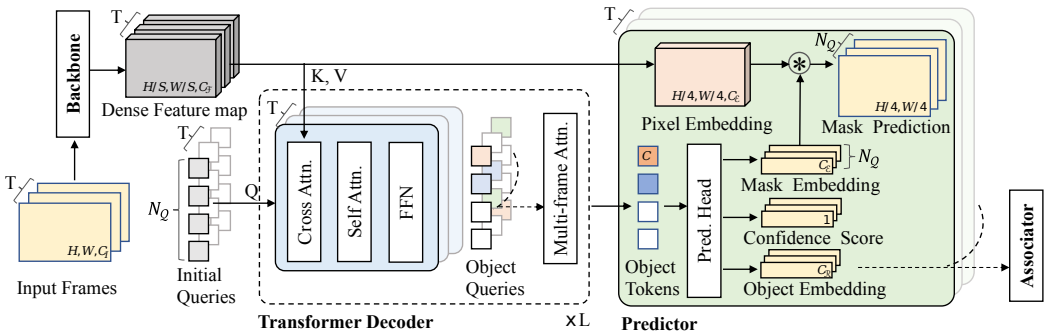

Figure 3: Outlined by a dashed rectangle, our transformer decoder ingests dense feature maps converted from input frames and produces object tokens for each image. These tokens predict confidence scores, mask embeddings for mask prediction, and object embeddings for association. We also introduce a novel "multi-frame attention" layer, which attends to object queries from all frames.

attention layer attends to object queries from all accessible frames. To illustrate, let us denote the intermediate object queries after each feed-forward network as $\mathbf{X}_l^t$, where l and t indicate the index of transformer blocks and frames, respectively. A standard self-attention layer (with a residual path) computes the following (we omit the normalization term $\sqrt{d_k}$ here for simplicity):

$$\text{SelfAttn}(\mathbf{X}_l^t) = \text{softmax}(f_Q(\mathbf{X}_l^t) \cdot f_K(\mathbf{X}_l^t)^T) f_V(\mathbf{X}_l^t) + \mathbf{X}_l^t \tag{3}$$

In contrast, our multi-frame attention layer computes:

$$\text{MultiFrameAttn}(\mathbf{X}_l^t) = \text{softmax}(f_Q(\mathbf{X}_l^t) \cdot f_K(\mathbf{X}_l)^T) f_V(\mathbf{X}_l) + \mathbf{X}_l^t \tag{4}$$

Here, $\mathbf{X}_l$ represents the set of object queries from all frames $\mathbf{X}_l = \{\mathbf{X}_l^t, \text{ for } t = 1, 2, \ldots, T\}$. The functions f_Q, f_K, and f_V are linear transformations that convert $\mathbf{X}_l$ or $\mathbf{X}_l^t$ into a C-dim space.

The multi-frame attention layer is positioned at the end of each transformer decoder block, which will be repeated L times in our network. That is to say, the output object queries are able to incorporate the dense feature map of the current frame to update its prediction after communicating with object queries from other frames. Furthermore, the multi-frame attention layer is computationally efficient as it only attends to the object queries from all frames, typically around 100 queries per frame, as opposed to 1200 queries per frame as in Mask2Former-video [12] using images with size 640×480. This efficiency allows it to process long-term input sequences and large-scale images.

4.4 Training and Losses

In our model, we adopt the same loss function as utilized in Mask2Former [24] for classification and mask prediction. This includes the softmax cross-entropy loss, denoted as $\mathcal{L}_{\text{class}}$, for classification, along with binary cross-entropy loss, denoted as $\mathcal{L}_{\text{ce}}$, and the Dice loss, denoted as $\mathcal{L}_{\text{dice}}$, for mask prediction. Concerning the loss for tracking, we employ the contrastive loss $\mathcal{L}_{\text{contra}}$ as used in DCN [29], along with the softmax loss (also referred to as the n-pair loss and InfoNCE loss) from CLIP [30]. For a object token $q_{\hat{t}}^{o_i}$, object tokens paired with the same object o_i in a different frames $q_t^{o_i}, t \neq \hat{t}$ treated as positive samples and pushed closer, while tokens assigned to a different objects or backgrounds are treated as negative pairs and pushed away. More details are available in the supplementary material..

Consistent with the approach in DETR [26] and Mask2Former [24], we leverage the Hungarian algorithm [28] to establish a bipartite matching between the predicted object tokens and ground truth that minimizes the overall loss. Notably, we exclude the tracking loss from the Hungarian algorithm's computation, given that $\mathcal{L}_{softmax}$ is influenced by the object tokens contributing to this loss. The final loss is the following:

$$\mathcal{L}_{\text{total}} = \lambda_{class}\mathcal{L}_{\text{class}} + \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{dice}\mathcal{L}_{\text{dice}} + \lambda_{\text{contra}}\mathcal{L}_{\text{contra}} + \lambda_{\text{softmax}}\mathcal{L}_{\text{softmax}}$$

Method	Shelf		Tabletop	
	AP@all	AP@0.5	AP@all	AP@0.5
MinVIS[13]	6.3	21.2	0.7	0.0
Mask2Former Video[12]	35.0	66.1	27.7	56.7
VITA[11]	42.7	70.1	26.6	55.0
STOW (Ours)	**55.6**	**81.3**	**49.7**	**75.4**

Table 1: Comparison between our method and state-of-the-art video instance segmentation methods. The networks are trained on synthetic data and evaluation is conducted on real data with no objects included in the training set. All of these experiments use ResNet-50[25] as backbone and witnessed same amount of images during training.

5 Experiments

We assess our methodology in two typical environments: bin in a shelf and tabletop, both of which are representative settings in a multitude of warehouse and domestic applications. For each setting, we generated corresponding synthetic data for the training phase and collected and annotated real-world data for evaluation. Our approach was also integrated into a bin-picking robotic system for practical experimentation.

Given the significant disparities between the shelf and table environments, we trained our models separately on the two distinct synthetic datasets. Each experimental run exclusively utilized one of these datasets and was evaluated against the corresponding real-world test set. The inference process requires approximately 0.4 seconds for a sequence of 15 frames using an RTX 2080Ti. Additional details pertaining to the training process are available in the supplementary material.

5.1 Dataset and Evaluation

Synthetic data. We constructed mesh models for both shelf and table bins, utilizing textures from the CC0 textures dataset [31]. For the object meshes, we use models from Google Scanned dataset [32] which contains more than 1000 models. We exclude 70 object meshes that have multiple isolated parts and set aside 100 objects to form the validation set. Roughly 900 objects are used to generate the training set. During the generation process, objects were randomly rotated along the z-axis and positioned to avoid collision with existing objects. For the shelf environment, we did not currently account for heavy occlusion; hence, no objects were placed behind others. In total, we generated around 10,000 sequences for the shelf environment, with each sequence including at least 2 fully packed frames, and approximately 2,000 sequences for the tabletop environment, with each sequence containing 15 frames.

Real-world data. For real-world scenarios, We collect and manually labeled 44 sequences with 220 images for the shelf scenario and 20 sequences with 280 images for the tabletop scenario. Our dataset includes a diverse variety of objects, with over 150 different kinds used in the data collection process. These ranged from relatively simple objects such as boxes and bottles to more challenging ones like transparent water bottles enclosed in plastic bags. In each sequence, we progressively added objects until either the bin was full or around 10 objects had been placed on the table. Object rearrangement could occur between any two frames, leading to significant changes in object location and appearance.

Evaluation metrics. We adopt the evaluation method of the VIS challenge [33], which is a modified version of the MS-COCO metric [34]. In video instance segmentation, each object is represented by a series of masks, and the Intersection over Union (IoU) is calculated at the level of these mask sequences. To construct the Precision-Recall (PR) curve, the confidence threshold is systematically varied, with each threshold yielding a distinct data point on the curve. The area under the PR curve provides the Average Precision (AP). In the context of this study, if not further specified, AP@0.5 denotes the average precision at an IoU threshold of 0.5. Similarly, AP@all represents the mean AP calculated over multiple IoU thresholds, specifically from 50% to 95% in 5% increments.

5.2 Results and Analysis

Baseline methods. We benchmark our approach against three state-of-the-art Video Instance Segmentation (VIS) methods: MinVIS [13], Mask2Former-video [12], and VITA [11]. To ensure an equitable comparison, all methods utilize ResNet-50 as the backbone, and the hyperparameters (such

as batch size, maximum iterations, and the number of sampled frames) are standardized to match ours. Hence, all methods are trained with an identical amount of images.

Qualitative results. As depicted in Table 1, our method notably outperforms the existing VIS techniques, yielding an approximate 10% improvement in the shelf environment and a 20% increase in the table environment, even without the utilization of the multi-frame attention layer. Interestingly, MinVIS exhibits subpar performance in our task. We speculate that this is due to its reliance on the proximity between object tokens as a measure of similarity. This method presupposes that the changes between frames are minimal and that objects maintain similar locations and appearances. However, these conditions do not consistently hold in our task, potentially explaining the lower performance of MinVIS.

Quantative results. We have also illustrated our results on a subset of real test data in Figure 4 and Figure 5. From these visualization results, it is evident that our approach adeptly handles frame changes amid various types of noise. Despite significant movement and rotation between frames, the network successfully segments and tracks a broad array of object categories. Whenever a new object is introduced, it efficiently segments and continues tracking it. Figure 5 also demonstrates the method's robustness against backgrounds, as it avoids predicting them as objects, even though walls were not included in the training set. Supplementary materials contain more results that contrast our approach with the other methods.

multi frame	shelf		tabletop	
	AP@all	AP@0.5	AP@all	AP@0.5
-	51.8	78.7	44.4	68.5
✓	**55.6**	**81.3**	**49.7**	**75.4**

Table 2: With and without the frame attention. The column on the left denoting whether we incorporate multi-frame attention layer in this experiemnts. All other hyper-parameters remain the same. We can see that after leveraging the frame attention layer, we can have a $\sim$5% boost both on shelf environment and tabletop environment.

5.3 Ablation Study

Frame attention. We evaluate the performance of our method with and without the cross-frame attention module on both the self and table scenes using ResNet-50 and Swin-T backbones. The cross-frame attention module brings consistent improvements to performance across all configurations, as shown in Table 2.

Sim2Real gap. We evaluated our methods on the synthetic validation set which consists of objects not included in the training set, and compare it with the number we tested on the real test set. Reseults are shown in Table 3. From the table, we can find that our counterparts get relatively compareable results than ours in the synthetic set, but their performance drops drastically when evaluate on the test set, which implies they have difficulties to solve the sim2real gap.

method	synthetic		real	
	AP@all	AP@0.5	AP@all	AP@0.5
MinVIS[13]	0.3	2.6	0.7	0.0
M2F-V[12]	71.6	83.7	27.7	56.7
VITA[11]	69.4	81.9	26.6	55.0
Ours	**74.1**	**89.3**	**45.5**	**71.9**

Table 3: Ablation Study on Solving sim2real gap. After training on synthetic tabletop training set, we evaluate each methods on synthetic tabletop validation set and real tabletop test set separately. Noticing that objects in synthetic validation set are not included in the synthetic training set.

5.4 Real-Robot Applications

Our visual perception technique has been integrated into an autonomous shelf-picking system. The system's software architecture is managed by a state machine and comprises several components. To assess the impact of our visual perception method on the system's performance, we designed an experiment with two test cases differing mainly in bin size and object types.

The system setup involves a UR16e industrial robot situated in front of an industrial warehouse shelf, laden with objects. We utilize UCN [4]+SIFT [7] as our baseline for comparison. Our evaluation protocol employs a fixed set of diverse items, stowed at specific locations and orientations within the bins to ensure reproducibility and comparability of results, as performance can fluctuate with different item configurations and inherent system stochasticity.

Figure 4: Visualized results on shelf environment. Masks with the same color and index denote being associated and predicted as the same object by the network

Figure 5: Visualized results on table environment. Masks with the same color and index denote being associated and predicted as the same object by the network

For the first test case, we use 14 objects, while the second test case involves 16 objects. In both scenarios, the objects are selected from 16 fully-loaded bins. Our novel method, designed for simultaneous object instance segmentation and tracking of unseen objects, substantially increased the successful pick rates. The success rate for the first test case rose from 36% to 52%, and for the second case, it improved from 27% to 67% - indicating a 1.4-fold improvement in the first scenario and an impressive 2.48-fold enhancement in the second.

6 Conclusion

In this paper, we define and address the task of segmenting and tracking unseen objects in discrete frames. We curate both synthetic and real datasets and present an innovative approach which successfully segments and tracks unseen objects in this context. This novel method adopts a new paradigm to do joint segmentation and tracking and introduce frame attention for efficient communication between frames. Despite being trained exclusively on synthetic data, our network capably segments and tracks objects amidst heavy clustering and large object movements within real-world image sequences. Our method surpasses existing state-of-the-art techniques on real-world datasets and real robot experiments. Extensive evaluations reveal that we consistently outperform leading methodologies in challenging real-world situations, achieving an improvement of 10% - 20% in AP.

Future work. Our method, while effective, still encounters difficulties with accurately segmenting objects in highly cluttered environments. Inspired by the notable results from SAM [6], we aim to incorporate it into our framework in future iterations. Additionaly, we will try explore learning-based algorithms to replace the current Hungarian algorithm-based associator.

References

[1] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox. Ifor: Iterative flow minimization for robotic object rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14787–14797, 2022.

[2] Y. Lu, N. Khargonkar, Z. Xu, C. Averill, K. Palanisamy, K. Hang, Y. Guo, N. Ruozzi, and Y. Xiang. Self-supervised unseen object instance segmentation via long-term robot interaction. *arXiv preprint arXiv:2302.03793*, 2023.

[3] C. Xie, Y. Xiang, A. Mousavian, and D. Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 37(5):1343–1359, 2021.

[4] Y. Xiang, C. Xie, A. Mousavian, and D. Fox. Learning rgb-d feature embeddings for unseen object instance segmentation. In *Conference on Robot Learning*, pages 461–470. PMLR, 2021.

[5] Y. Lu, Y. Chen, N. Ruozzi, and Y. Xiang. Mean shift mask transformer for unseen object instance segmentation. *arXiv preprint arXiv:2211.11679*, 2022.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[8] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[9] H. K. Cheng and A. G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 640–658. Springer, 2022.

[10] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos, 2023.

[11] M. Heo, S. Hwang, S. W. Oh, J.-Y. Lee, and S. J. Kim. Vita: Video instance segmentation via object token association. *arXiv preprint arXiv:2206.04403*, 2022.

[12] B. Cheng, A. Choudhuri, I. Misra, A. Kirillov, R. Girdhar, and A. G. Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

[13] D.-A. Huang, Z. Yu, and A. Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245*, 2022.

[14] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019.

[15] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[16] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468, 2016.

[18] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[19] J. Luiten, I. E. Zulfikar, and B. Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2000–2009, 2020.

[20] S. Garg and V. Goel. Mask selection and propagation for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1680–1690, 2021.

[21] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[22] A. Wu, W.-S. Zheng, H.-X. Yu, S. Gong, and J. Lai. Rgb-infrared cross-modality person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 5380–5389, 2017.

[23] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 993–1000. IEEE, 2006.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[27] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[28] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[29] P. R. Florence. *Dense visual learning for robot manipulation*. PhD thesis, Massachusetts Institute of Technology, 2020.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[31] ambientCG. Creative Commons Zero (CC0). https://ambientcg.com, 2023.

[32] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[33] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.