# Interactive Video Object Segmentation Using Global and Local Transfer Modules

Yuk Heo[1][0000−0002−7425−1254], Yeong Jun Koh[2][0000−0003−1805−2960], and Chang-Su Kim[1][0000−0002−4276−1831]

[1] School of Electrical Engineering, Korea University, Korea
yukheo@mcl.korea.ac.kr changsukim@korea.ac.kr
[2] Department of Computer Science & Engineering, Chungnam National University, Korea
yjkoh@cnu.ac.kr

**Abstract.** An interactive video object segmentation algorithm, which takes scribble annotations on query objects as input, is proposed in this paper. We develop a deep neural network, which consists of the annotation network (A-Net) and the transfer network (T-Net). First, given user scribbles on a frame, A-Net yields a segmentation result based on the encoder-decoder architecture. Second, T-Net transfers the segmentation result bidirectionally to the other frames, by employing the global and local transfer modules. The global transfer module conveys the segmentation information in an annotated frame to a target frame, while the local transfer module propagates the segmentation information in a temporally adjacent frame to the target frame. By applying A-Net and T-Net alternately, a user can obtain desired segmentation results with minimal efforts. We train the entire network in two stages, by emulating user scribbles and employing an auxiliary loss. Experimental results demonstrate that the proposed interactive video object segmentation algorithm outperforms the state-of-the-art conventional algorithms. Codes and models are available at https://github.com/yuk6heo/IVOS-ATNet.

**Keywords:** Video object segmentation, interactive segmentation, deep learning

## 1 Introduction

Video object segmentation (VOS) aims at separating objects of interest from the background in a video sequence. It is an essential technique to facilitate many vision tasks, including action recognition, video retrieval, video summarization, and video editing. Many researches have been carried out to perform VOS, and it can be categorized according to the level of automation. Unsupervised VOS segments out objects with no user annotations, but it may fail to detect objects of interest or separate multiple objects. Semi-supervised VOS extracts target objects, which are manually annotated by a user in the first frame or only a few frames in a video sequence. However, semi-supervised approaches require time-consuming pixel-level annotations (at least 79 seconds per instance as revealed in [5]) to delineate objects of interest.

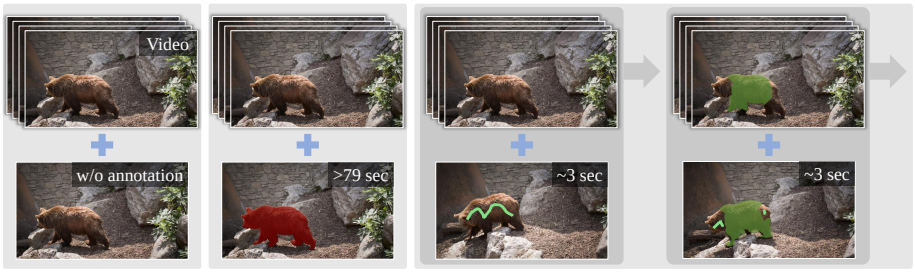


Fig. 1: Three different levels of supervision in (a) unsupervised VOS, (b) semi-supervised VOS, and (c) interactive VOS. Unsupervised VOS demands no user interaction. Semi-supervised VOS needs pixel-level annotations of an object. Interactive VOS uses quick scribbles and allows interactions with a user repeatedly.

Therefore, as an alternative approach, we consider interactive VOS, which allows users to interact with segmentation results repeatedly using simple annotations, *e.g.* scribbles, point clicks, or bounding boxes. In this regard, the objective of interactive VOS is to provide reliable segmentation results with minimal user efforts. A work-flow to achieve this objective was presented in the 2018 DAVIS Challenge [5]. This work-flow employs scribble annotations as supervision, since it takes only about 3 seconds to draw a scribble on an object instance. In this scenario, a user provides scribbles on query objects in a selected frame and the VOS algorithm yields segment tracks for the objects in all frames. We refer to this turn of user-algorithm interaction as a segmentation round. Then, we repeat segmentation rounds to refine the segmentation results until satisfactory results are obtained as illustrated in Fig. 1(c).

In this paper, we propose a novel approach to achieve interactive VOS using scribble annotations with the work-flow in [5]. First, we develop the annotation network (A-Net), which produces a segmentation mask for an annotated frame using scribble annotations for query objects. Next, we propose the transfer network (T-Net) to transfer the segmentation result to other target frames subsequently to obtain segment tracks for the query objects. We design the global transfer module and the local transfer module in T-Net to convey segmentation information reliably and accurately. We train A-Net and T-Net in two stages by mimicking scribbles and employing an auxiliary loss. Experimental results verify that the proposed algorithm outperforms the state-of-the-art interactive VOS algorithms on the DAVIS2017 [35]. Also, we perform a user study to demonstrate the effectiveness of the proposed algorithm in real-world applications.

This paper has three main contributions:

1. Architecture of A-Net and T-Net with the global and local transfer modules.
2. Training strategy with the scribble imitation and the auxiliary loss to activate the local transfer module and make it effective in T-Net.
3. Remarkable performance on the DAVIS dataset in various conditions.

## 2 Related Work

**Unsupervised VOS:** Unsupervised VOS is a task to segment out primary objects [22] in a video without any manual annotations. Before the advance of deep learning, diverse information, including motion, object proposals, and saliency, was employed to solve this problem [21, 23, 24, 32, 47]. Recently, many deep learning algorithms with different network architectures have been developed to improve VOS performance using big datasets [35, 53]. Tokmakov *et al*. [43] presented a fully convolutional model to learn motion patterns from videos. Jain *et al*. [17] merged appearance and motion information to perform unsupervised segmentation. Song *et al*. [41] proposed an algorithm using LSTM architecture [11] with atrous convolution layers [6]. Wang *et al*. [48] also adopted LSTM with a visual attention module to simulate human attention.

**Semi-supervised VOS:** Semi-supervised VOS extracts query objects using accurately annotated masks at the first frames. Early methods for semi-supervised VOS were developed using hand-crafted features based on random walkers, trajectories, or super-pixels [3, 16, 18]. Recently, deep neural networks have been adopted for semi-supervised VOS. Some deep learning techniques [4, 26, 45] are based on a time-consuming online learning, which fine-tunes a pre-trained network using query object masks at the first frame. Without the fine-tuning, the algorithms in [9,19,29,33,54] propagate segmentation masks, which are estimated in the previous frame, to the current target frame sequentially for segmenting out query objects. Jang *et al*. [19] warped segmentation masks in the previous frame to the target frame and refined the warped masks through convolution trident networks. Yang *et al*. [54] encoded object location information from a previous frame and combined it with visual appearance features to segment out the query object in the target frame. Also, the algorithms in [8,15,31,44] perform matching between the first frame and a target frame in an embedding space to localize query objects. Chen *et al*. [8] dichotomized each pixel into either object or background using features from the embedding network. Voigtlaender *et al*. [44] trained their embedding network to perform the global and local matching.

**Interactive image segmentation:** Interactive image segmentation aims at extracting a target object from the background using user annotations. As annotations, bounding boxes were widely adopted in early methods [25,38,42,50]. Recently, point-interfaced techniques have been developed [20, 27, 40, 52]. Maninis *et al*. [27] used four extreme points as annotations to inform their network about object boundaries. Jang and Kim [20] corrected mislabeled pixels through the backpropagating refinement scheme.

**Interactive VOS:** Interactive VOS allows users to interact with segmentation results repeatedly using various input types, *e.g*. points, scribbles, and bounding boxes. Users can refine segmentation results until they are satisfied. Some interactive VOS algorithms [36,39,46] build graph models using the information in user strokes and segment out target objects via optimization. In [1,10], patch matching between target and reference frames is performed to localize query objects. Box interactions can be provided to correct box positions. Benard and Gygli [2] employed two deep learning networks to achieve interactive VOS. They

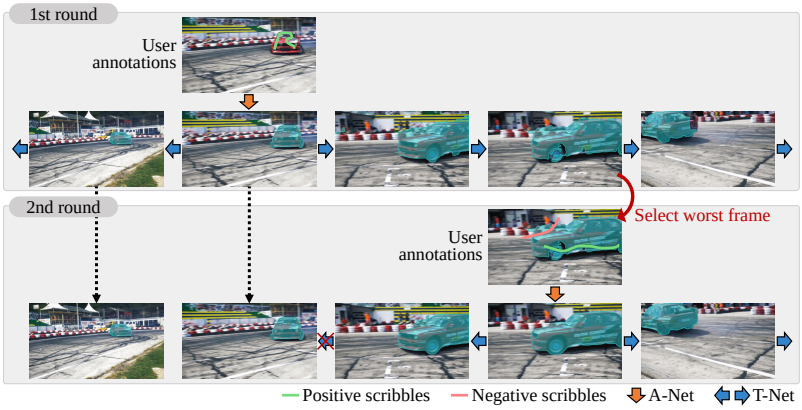


Fig. 2: Overview of the proposed interactive VOS algorithm.

first obtained object masks from point clicks or scribbles using an interactive image segmentation network and then segmented out the objects using a semi-supervised VOS network. Chen *et al*. [8] demanded only a small number of point clicks based on pixel-wise metric learning. Oh *et al*. [30] achieved interactive VOS by following the work-flow in [5]. They used two segmentation networks to obtain segmentation masks from user scribbles and to propagate the segmentation masks to neighboring frames by exploiting regions of interest. However, their networks may fail to extract query objects outside the regions of interest.

## 3 Proposed Algorithm

We segment out one or more objects in a sequence of video frames through user interactions. To this end, we develop two networks: 1) annotation network (A-Net) and 2) transfer network (T-Net).

Fig. 2 is an overview of the proposed algorithm. In the first segmentation round, a user provides annotations (*i.e*. scribbles) for a query object to A-Net, which then yields a segmentation mask for the annotated frame. Then, T-Net transfers the segmentation mask bi-directionally to both ends of the video to compose a segment track for the object. From the second round, the user selects the poorest segmented frame, and then provides positive and negative scribbles so that A-Net corrects the result. Then, T-Net again propagates the refined segmentation mask to other frames until a previously annotated frame is met. This process is repeated until satisfactory results are obtained.

### 3.1 Network architecture

Fig. 3 shows the architecture of the proposed algorithm, which is composed of A-Net and T-Net. First, we segment out query objects in an annotated frame $I_a$ via A-Net. Then, to achieve segmentation in a target frame $I_t$, we develop T-Net, which includes the global and local transfer modules.

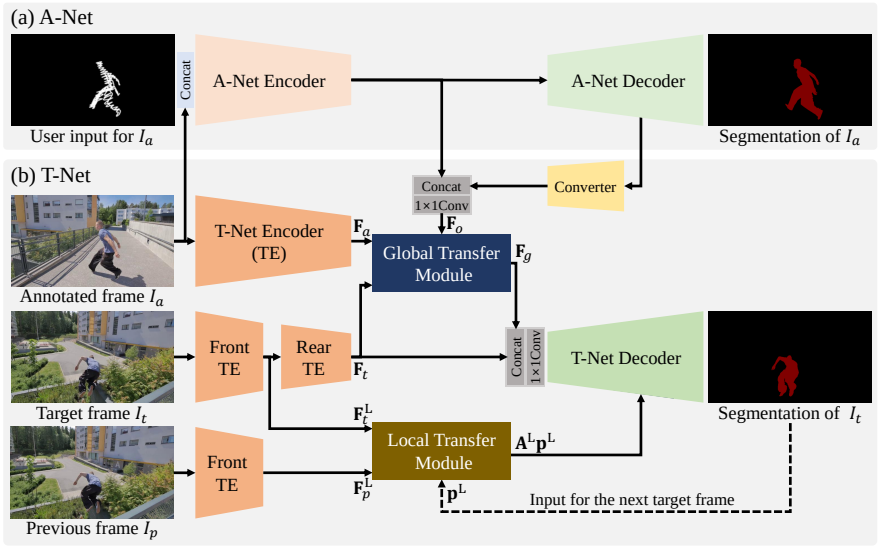


Fig. 3: Architecture of the proposed networks. A target object in an annotated frame $I_a$ is extracted by A-Net in (a), and the result is sequentially propagated to the other frames, called target frames, by T-Net in (b). In this diagram, skip connections are omitted.

**A-Net:** Through user interactions, A-Net infers segmentation results in an annotated frame $I_a$. There are two types of interactions according to iteration rounds. In the first round, a user draws scribbles on target objects. In this case, A-Net accepts four-channel input: RGB channels of $I_a$ and one scribble map. In subsequent rounds, the user supplies both positive and negative scribbles after examining the segmentation results in the previous rounds, as illustrated in Fig. 2. Hence, A-Net takes six channels: RGB channels, segmentation mask map in the previous round, and positive and negative scribble maps. We design A-Net to take 6-channel input, but in the first round, fill in the segmentation mask map with 0.5 and the negative scribble map with 0.

A-Net has the encoder-decoder architecture, as specified in Fig. 4(a). We adopt SE-ResNet50 [14] as the encoder to extract features and employ skip connections to consider both low-level and high-level features. We perform dilated convolution and exclude max-pooling in the R5 convolution layer. Then, we use two parallel modules: an ASPP module [7], followed by up-sampling with bilinear interpolation, and a bottom-up module. ASPP analyzes multi-scale context features using dilated convolution with varying rates. The bottom-up module consists of two refine modules [29]. The output signals of the ASPP and bottom-up modules are concatenated and then used to predict a probability map of a query object through three sets of convolutional layers, ReLU, and batch normalization. Finally, the estimated probability map is up-sampled to be of the same size as the input image using bilinear interpolation.

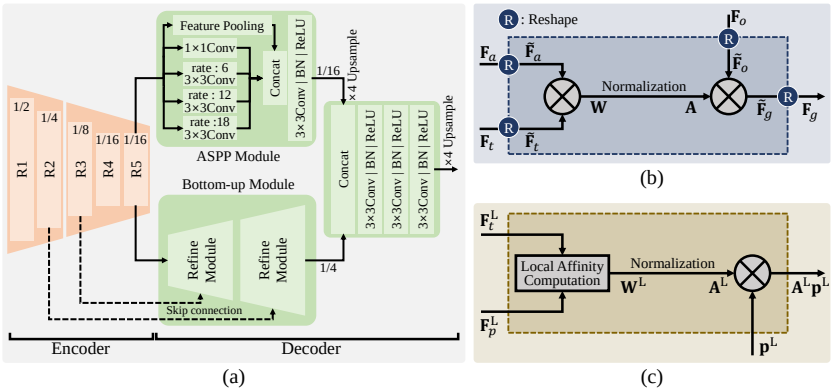


Fig. 4: (a) The encoder-decoder architecture, adopted by the proposed A-Net and T-Net. Each fraction is the ratio of the output feature resolution to the input image resolution. (b) Global transfer module. (c) Local transfer module.

**T-Net:** We develop T-Net, which consists of shared encoders, a global transfer module, a local transfer module, and a decoder, as shown in Fig. 3(b). The encoders and decoder in T-Net have the same structures as those of A-Net in Fig. 4(a). The T-Net decoder yields a probability map for query objects in a target frame $I_t$ using the features from the encoder, the global transfer module, and the local transfer module. Let us describe these two transfer modules.

**Global transfer module:** We design the global transfer module to convey the segmentation information of the annotated frame $I_a$ to the target frame $I_t$. Fig. 4(b) shows its structure, which adopts the non-local model in [49]. It takes two feature volumes $\mathbf{F}_t$ and $\mathbf{F}_a$ for $I_t$ and $I_a$, respectively. Each volume contains $C$-dimensional feature vectors for $H \times W$ pixels. We then construct an affinity matrix $\mathbf{W}$ between $I_t$ and $I_a$, by computing the inner products between all possible pairs of feature vectors in $\mathbf{F}_t$ and $\mathbf{F}_a$. Specifically, let $\tilde{\mathbf{F}}_t \in \mathbb{R}^{HW \times C}$ and $\tilde{\mathbf{F}}_a \in \mathbb{R}^{HW \times C}$ denote the feature volumes reshaped into matrices. We perform the matrix multiplication to obtain the affinity matrix

$$\mathbf{W} = \tilde{\mathbf{F}}_t \times \tilde{\mathbf{F}}_a^T. \tag{1}$$

Its element $\mathbf{W}(i, j)$ represents the affinity of the $i$th pixel in $\tilde{\mathbf{F}}_t$ to the $j$th pixel in $\tilde{\mathbf{F}}_a$. Then, we obtain the transition matrix $\mathbf{A}$ by applying the softmax normalization to each column in $\mathbf{W}$.

The transition matrix $\mathbf{A}$ contains matching probabilities from pixels in $I_a$ to those in $I_t$. Therefore, it can transfer query object probabilities in $I_a$ to $I_t$. To approximate these probabilities in $I_a$, we extract a mid-layer feature from the A-Net decoder, down-sample it using the converter, which includes two sets of SE-Resblock [14] and max-pooling layer. Then, its channels are halved by 1×1 convolutions after it is concatenated to the output of the A-Net encoder,

as shown in Fig. 3. The concatenated feature $\mathbf{F}_o$ is fed into the global transfer module, as shown in Fig. 4(b). Then, it is reshaped into $\tilde{\mathbf{F}}_o$, which represents the query object feature distribution in $I_a$. Finally, the global transfer module produces the transferred distribution

$$\tilde{\mathbf{F}}_g = \mathbf{A}\tilde{\mathbf{F}}_o, \tag{2}$$

which can be regarded as an inter-image estimate of the query object feature distribution in $I_t$. Then the distribution is reshaped into $\mathbf{F}_g \in \mathbb{R}^{H\times W\times C}$ to be input to the T-Net decoder.

From the second round, there are $N$ annotated frames, where $N$ is the ordinal index for the round. To obtain reliable segmentation results, we use all information in the $N$ annotated frames. Specifically, we compute the transition matrix $\mathbf{A}^{(i)}$ from the $i$th annotated frame to $I_t$ and the query object distribution $\mathbf{F}_o^{(i)}$ in the $i$th annotated frame. Then, we obtain the average of the multiple inter-image estimates of the query object distribution in $I_t$ by

$$\tilde{\mathbf{F}}_g = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}^{(i)}\tilde{\mathbf{F}}_o^{(i)}. \tag{3}$$

**Local transfer module:** The segmentation information in an annotated frame is propagated bidirectionally throughout the sequence. Thus, during the propagation, when a target frame $I_t$ is to be segmented, there is the previous frame $I_p$ that is already segmented. We design the local transfer module to convey the segmentation information in $I_p$ to $I_t$.

The local transfer module is similar to the global one, but it performs matching locally since $I_t$ and $I_p$ are temporally adjacent. In other words, object motions between $I_t$ and $I_p$, which tend to be smaller than those between $I_t$ and $I_a$, are estimated locally. Furthermore, since $I_t$ and $I_p$ are more highly correlated, motions between them can be estimated more accurately. Therefore, the local module uses higher-resolution features than the global one does. Specifically, the local module takes features from the R2 convolution layer in the encoder in Fig. 4(a), instead of the R5 layer. $\mathbf{F}_t^{\mathrm{L}}$ and $\mathbf{F}_p^{\mathrm{L}}$, which denote these feature volumes from $I_t$ and $I_p$, are provided to the local transfer module, as shown in Fig. 4(c). Then, we compute the local affinity matrix $\mathbf{W}^{\mathrm{L}}$, whose $(i,j)$th element indicates the similarity between the $i$th pixel $I_t$ and the $j$th pixel in $I_p$. Specifically, $\mathbf{W}^{\mathrm{L}}(i,j)$ is defined as

$$\mathbf{W}^{\mathrm{L}}(i,j) = \begin{cases} \mathbf{f}_{t,i}^T\mathbf{f}_{p,j} & j \in \mathcal{N}_i, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathbf{f}_{t,i}$ and $\mathbf{f}_{p,j}$ are the feature vectors for the $i$th pixel in $\mathbf{F}_t^{\mathrm{L}}$ and the $j$th pixel in $\mathbf{F}_p^{\mathrm{L}}$, respectively. Also, the local region $\mathcal{N}_i$ is the set of pixels, which are sampled from $(2d+1)\times(2d+1)$ pixels around pixel $i$ with stride 2 to reduce the computations. In this work, $d$ is set to 4. Then, the affinity is computed for those pixels in the local region only, and set to be zeros for the other pixels.

As in the global module, $\mathbf{W}^{\mathrm{L}}$ is normalized column-by-column to the transition matrix $\mathbf{A}^{\mathrm{L}}$. Also, a segmentation mask map $\mathbf{P}_p$ in the previous frame $I_p$ is down-sampled and vectorized to obtain a probability vector $\mathbf{p}^{\mathrm{L}}$. Then, we obtain $\mathbf{A}^{\mathrm{L}}\mathbf{p}^{\mathrm{L}}$, which is another estimate of the query object distribution in $I_t$. It has a higher resolution than the estimate in the global module, and thus is added to the corresponding mid-layer in the T-Net decoder, as shown in Fig. 3(b).

Computing global and local similarities in the proposed global and local transfer modules is conceptually similar to [44], but their usage is significantly different. Although [44] also computes global and local distances, it transforms those distances into a single channel by taking the minimum distance at each position. Thus, it loses a substantial amount of distance information. In contrast, the proposed algorithm computes global and local affinity matrices and uses them to transfer object probabilities from annotated and previous frames to a target frame. In Section 4.3, we verify that the proposed global and local modules are more effective than the best matching approach in [44].

### 3.2 Training phase

We train the proposed interactive VOS networks in two stages, since T-Net should use A-Net output; we first train A-Net and then train T-Net using the trained A-Net.

**A-Net training:** To train A-Net, we use the image segmentation dataset in [12] and the video segmentation datasets in [35,53]. Only a small percentage of videos in the DAVIS2017 dataset [35] provide user scribble data. Hence, we emulate user scribbles via two schemes: 1) point generation and 2) scribble generation in [5].

In the first round, A-Net yields a segmentation mask for a query object using positive scribbles only. We perform the point generation to imitate those positive scribbles. We produce a point map by sampling points from the ground-truth mask for the query object. Specifically, we pick one point randomly for every $100 \sim 3000$ object pixels. We vary the sampling rate to reflect that users provide scribbles with different densities. Then, we use the generated point map as the positive scribble map.

In each subsequent round, A-Net should refine the segmentation mask in the previous round using both positive and negative scribbles. To mimic an inaccurate segmentation mask, we deform the ground-truth mask using various affine transformations. Then, we extract positive and negative scribbles using the scribble generation scheme in [5], by comparing the deformed mask with the ground-truth. Then, $I_a$, the deformed mask, and the generated positive and negative scribble maps are fed into A-Net for training.

We adopt the pixel-wise class-balanced cross-entropy loss [51] between A-Net output and the ground-truth. We adopt the Adam optimizer to minimize this loss for 60 epochs with a learning rate of $1.0 \times 10^{-5}$. We decrease the learning rate by a factor of 0.2 every 20 epochs. In each epoch, the training is iterated for 7,000 mini-batches, each of which includes 6 pairs of image and ground-truth. For data augmentation, we apply random affine transforms to the pairs.

**T-Net training:** For each video, we randomly pick one frame as an annotated frame, and then select seven consecutive frames, adjacent to the annotated frame, in either the forward or backward direction. Among those seven frames, we randomly choose four frames to form a mini-sequence. Thus, there are five frames in a mini-sequence: one annotated frame and four target frames. For each target frame in the mini-sequence, we train T-Net using the features from the trained A-Net, which takes the annotated frame as input.

We compare an estimated segmentation mask with the ground-truth to train T-Net, by employing the loss function

$$\mathcal{L} = \mathcal{L}_{\mathrm{c}} + \lambda \mathcal{L}_{\mathrm{aux}} \tag{5}$$

where $\mathcal{L}_{\mathrm{c}}$ is the pixel-wise class-balanced cross-entropy loss between the T-Net output and the ground-truth. The auxiliary loss $\mathcal{L}_{\mathrm{aux}}$ is the pixel-wise mean square loss between the transferred probability map, which is the output of the local transfer module, and the down-sampled ground-truth. The auxiliary loss $\mathcal{L}_{\mathrm{aux}}$ enforces the front encoders of T-Net in Fig. 3 to generate appropriate features for transferring the previous segmentation mask successfully. Also, $\lambda$ is a balancing hyper-parameter, which is set to 0.1. We also employ the Adam optimizer to minimize the loss function for 40 epochs with a learning rate of $1.0 \times 10^{-5}$, which is decreased by a factor of 0.2 every 20 epochs. The training is iterated 6,000 mini-batches, each of which contains 8 mini-sequences.

### 3.3 Inference phase

Suppose that there are multiple target objects. In the first round, for each target object in an annotated frame, A-Net accepts the user scribbles on the object and produces a probability map for the object. To obtain multiple object segmentation results, after zeroing probabilities lower than 0.8, each pixel is assigned to the target object class, corresponding to the highest probability. Then, T-Net transfers the multiple segmentation masks in the annotated frame bi-directionally to both ends of the sequence. T-Net also compares the multiple probability maps and determines the target object class of each pixel, as done in A-Net. From the second round, the user selects the frame with the poorest segmentation results and then provides additional positive and negative scribbles. The scribbles are then fed into A-Net to refine the segmentation results. Then, we transfer segmentation results bidirectionally with T-Net. In each direction, the transmission is carried out until another annotated frame is found.

During the transfer, we superpose the result of segmentation mask $\mathbf{P}_t^r$ for frame $I_t$ in the current round $r$ with that $\mathbf{P}_t^{r-1}$ in the previous round. Specifically, the updated result $\tilde{\mathbf{P}}_t^r$ in round $r$ is given by

$$\tilde{\mathbf{P}}_t^r = \frac{1}{2}\left(1 + \frac{t - t_b}{t_r - t_b}\right)\mathbf{P}_t^r + \frac{t_r - t}{2(t_r - t_b)}\mathbf{P}_t^{r-1} \tag{6}$$

where $t_r$ is the annotated frame in round $r$ and $t_b$ is one of the previously annotated frames, which is the closest to $t$ in the direction of the transfer. By

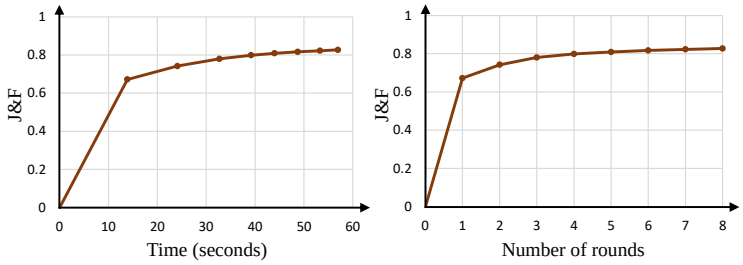


Fig. 5: J&F performances of the proposed algorithm on the validation set in DAVIS2017 according to the time and the number of rounds.

Table 1: Comparison of the proposed algorithm with the conventional algorithms on the DAVIS2017 validation set. The best results are boldfaced.

| | AUC-J | J@60s | AUC-J&F | J&F@60s |
|---|---|---|---|---|
| Najafi *et al.* [28] | 0.702 | 0.548 | – | – |
| Heo *et al.* [13] | 0.704 | 0.725 | 0.734 | 0.752 |
| Ren *et al.* [37] | – | – | 0.766 | 0.780 |
| Oh *et al.* [30] | 0.691 | 0.734 | – | – |
| Proposed | **0.771** | **0.790** | **0.809** | **0.827** |

employing this superposition scheme, we can reduce drifts due to a long temporal distance between annotated and target frames.

## 4 Experimental Results

We first compare the proposed interactive VOS algorithm with conventional algorithms. Second, we conduct a user study to assess the proposed algorithm in real-world applications. Finally, we do various ablation studies to analyze the proposed algorithm.

### 4.1 Comparative assessment

In this test, we follow the interactive VOS work-flow in [5]. The work-flow first provides a manually generated scribble for each target object in the first round, and then automatically generates additional positive and negative scribbles to refine the worst frames in up to 8 subsequent rounds. There are three different scribbles provided in the first round. In other words, three experiments are performed for each video sequence. The region similarity (J) and contour accuracy (F) metrics are employed to assess VOS algorithms. For the evaluation of interactive VOS, we measure the area under the curve for J score (AUC-J) and for joint J and F scores (AUC-J&F) to observe the overall performance according

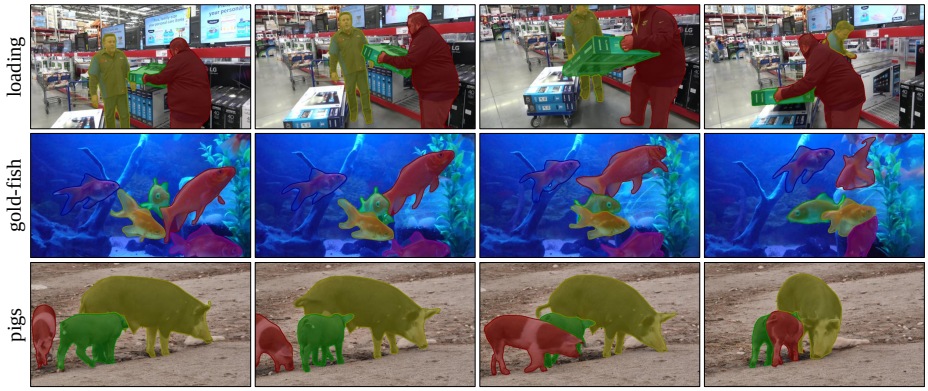


Fig. 6: Results of the proposed interactive VOS algorithm after 8 rounds.

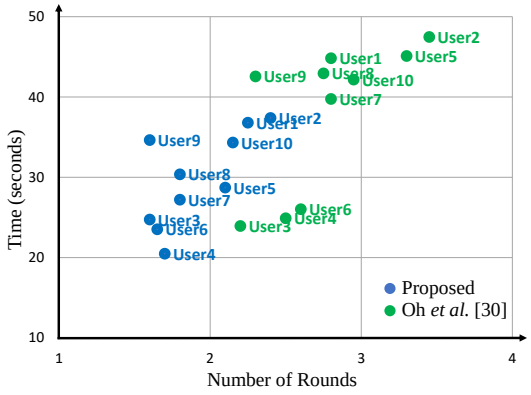


Fig. 7: Comparison of the average times and average round numbers.

over the 8 segmentation rounds. Also, we measure the J score at 60 seconds (J@60s), and the joint J and F score at 60 seconds (J&F@60s) to evaluate how much performance is achieved within the restricted time.

Fig. 5 shows the J&F performances of the proposed algorithm on the validation set in DAVIS2017 [35] according to the time and the number of rounds. The performances increase quickly and saturate at around 40s or in the third round. Also, we observe that the 8-round experiment is completed within 60 seconds. Table 1 compares the proposed algorithm with recent state-of-the-art algorithms [13, 28, 30, 37]. The scores of the conventional algorithms are from the respective papers. The proposed algorithm outperforms the conventional algorithms by significant margins in all metrics. Fig. 6 presents examples of segmentation results of the proposed algorithm after 8 rounds. We see that multiple primary objects are segmented out faithfully.

Table 2: Summary of the user study results.

| | SPV | RPV | J Mean | F Mean |
|---|---|---|---|---|
| Oh *et al.* [30] | 37.9 | 2.77 | 0.823 | 0.817 |
| Proposed | **29.8** | **1.90** | **0.832** | **0.822** |


Fig. 8: Examples of scribbles and segmentation results during the user study. Positive and negative scribbles are depicted in green and red, respectively.

### 4.2 User study

We conduct a user study, by recruiting 10 off-line volunteers and asking them to provide scribbles repeatedly until they are satisfied. We measure the average time in seconds per video (SPV), including the interaction time to provide scribbles and the running time of the algorithm, and the average round number in rounds per video (RPV) until the completion. Also, we report the J and F means of all sequences when the interactive process is completed.

We perform the user study for the proposed algorithm and the state-of-the-art interactive VOS algorithm [30]. For this comparison, we use the validation set (20 sequences) in DAVIS2016 [34], in which each video contains only a single query object. This is because the provided source code of [30] works on a single-object case only. Fig. 7 plots the average time and the average round number for each user. We observe that all users, except user 3, spend less time and conduct fewer rounds using the proposed algorithm. Table 2 summarizes the user study results. The proposed algorithm is faster than [30] in terms of both SPV and RPV. It is worth pointing out that the proposed algorithm yields better segmentation results within shorter times.

Fig. 8 shows examples of segmentation results in the user study. For the "Libby," "Horsejump-High," and "Parkour" sequences, the proposed algorithm deals with occlusions and scale changes of query objects effectively, and completes the segmentation in just a single round. Please see the supplemental video to see how the evaluation works.

Table 3: Ablation study on the local transfer module (J scores on the validation set in DAVIS2017).

| Method | Front TE | Rear TE | $\lambda$ | Round 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|---|---|
| I | w/o local transfer module | | | 0.629 | 0.704 | 0.741 | 0.759 | 0.771 |
| II | | ✓ | 0.1 | 0.653 | 0.708 | 0.738 | 0.751 | 0.760 |
| III | ✓ | | 0.0 | 0.645 | 0.706 | 0.735 | 0.750 | 0.761 |
| IV | ✓ | | 0.5 | 0.658 | 0.721 | 0.748 | 0.758 | 0.772 |
| V | ✓ | | 1.0 | 0.654 | 0.715 | 0.742 | 0.755 | 0.762 |
| VI (Proposed) | ✓ | | 0.1 | **0.676** | **0.732** | **0.762** | **0.772** | **0.783** |

Table 4: Ablation study to validate the proposed probability transfer approach.

| | AUC-J | J@60s | AUC-J&F | J&F@60s |
|---|---|---|---|---|
| Matching approach [44] (predictions of A-Net) | 0.636 | 0.653 | 0.654 | 0.670 |
| Matching approach [44] (scribble annotations) | 0.661 | 0.676 | 0.674 | 0.690 |
| Proposed probability transfer approach | **0.771** | **0.790** | **0.809** | **0.827** |

### 4.3 Ablation studies

We analyze the efficacy of the proposed global and local transfer modules through two ablation studies.

First, we verify that the structure and the training method of the local transfer module are effective. In Table 3, we report the J scores on the validation set in DAVIS2017, by varying the configurations of the local transfer module. In method I, we assess the proposed algorithm without the local transfer module. Note that the J scores in early rounds degrade severely. The local model is hence essential for providing satisfactory results to users quickly in only a few rounds. Method II uses the features of rear TE, instead of those of front TE to compute the affinity matrix of the local transfer module. The features of the front TE are more effective than those of rear TE because of their higher spatial resolution. In method III, without the auxiliary loss $\mathcal{L}_{\text{aux}}$ (*i.e.* $\lambda = 0$ in (5)), the local transfer module becomes ineffective and the performances degrade significantly. Methods IV, V, and VI vary the parameter $\lambda$. We see that $\lambda = 0.1$ performs the best by balancing the two losses in (5).

Next, we verify that the proposed global and local transfer modules are more effective for interactive VOS than the global and local matching in [44]. Note that [44] is a semi-supervised VOS algorithm, which estimates matching maps between a target frame and the target object region. We plug its matching modules into the proposed interactive system. More specifically, we compute a global similarity map between a target frame and the target object region in an annotated frame to perform the global matching in [44]. We determine the target object region in two ways: 1) the region predicted by A-Net or 2) the

set of scribble-annotated pixels. We then transform the similarity map into a single channel by taking the maximum similarity at each position. Then, we replace $\mathbf{F}_g$, which is the output of the proposed global transfer module, with the single-channel similarity. For the local matching, we obtain a local similarity map between the target frame and the segmentation region in the previous frame to compose another single-channel similarity. We then feed the local matching result, instead of $\mathbf{A}^{\mathrm{L}}\mathbf{p}^{\mathrm{L}}$, to the T-Net decoder. We train these modified networks using the same training set as the proposed networks. The implementation details of the modified networks can be found in the supplemental document. Table 4 compares the performances of the proposed transfer modules with those of the matching modules in [44] on the validation set in DAVIS2017. We observe that the proposed probability transfer approach outperforms the best matching approach [44] significantly.

## 5 Conclusions

We proposed a novel interactive VOS algorithm using A-Net and T-Net. Based on the encoder-decoder architecture, A-Net processes user scribbles on an annotated frame to generate a segmentation result. Then, using the global and local transfer modules, T-Net conveys the segmentation information to the other frames in the video sequence. These two modules are complementary to each other. The global module transfers the information from an annotated frame to a target frame reliably, while the local module conveys the information between adjacent frames accurately. In the training process, we introduced the point-generation method to compensate for the lack of scribble-annotated data. Moreover, we incorporated the auxiliary loss to activate the local transfer module and make it effective in T-Net. By employing A-Net and T-Net repeatedly, a user can obtain satisfactory segmentation results. Experimental results showed that the proposed algorithm performs better than the state-of-the-art algorithms, while requiring fewer interaction rounds.

## Acknowledgements

This work was supported in part by 'The Cross-Ministry Giga KOREA Project' grant funded by the Korea government (MSIT) (No.GK20P0200, Development of 4D reconstruction and dynamic deformable action model based hyper-realistic service technology), in part by Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01441, Artificial Intelligence Convergence Research Center (Chungnam National University)) and in part by the National Research Foundation of Korea (NRF) through the Korea Government (MSIP) under Grant NRF-2018R1A2B3003896.

## References


1. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video SnapCut: robust video object cutout using localized classifiers. ACM Trans. on Graphics **28**(3), 70 (2009) 3
2. Benard, A., Gygli, M.: Interactive video object segmentation in the wild. arXiv:1801.00269 (2017) 3
3. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV (2010) 3
4. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017) 3
5. Caelles, S., Montes, A., Maninis, K.K., Chen, Y., Van Gool, L., Perazzi, F., Pont-Tuset, J.: The 2018 DAVIS challenge on video object segmentation. arXiv:1803.00557 (2018) 1, 2, 4, 8, 10
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv:1412.7062 (2014) 3
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018) 5
8. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR (2018) 3, 4
9. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: SegFlow: Joint learning for video object segmentation and optical flow. In: ICCV (2017) 3
10. Fan, Q., Zhong, F., Lischinski, D., Cohen-Or, D., Chen, B.: JumpCut: non-successive mask transfer and interpolation for video cutout. ACM Trans. on Graphics **34**(6), 195:1–195:10 (2015) 3
11. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. Neural Computation **12**(10), 2451–2471 (1999) 3
12. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011) 8
13. Heo, Y., Koh, Y.J., Kim, C.S.: Interactive video object segmentation using sparse-to-dense networks. In: CVPRW (2019) 10, 11
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018) 5, 6
15. Hu, Y.T., Huang, J.B., Schwing, A.G.: Videomatch: Matching based video object segmentation. In: ECCV (2018) 3
16. Jain, S.D., Grauman, K.: Supervoxel-consistent foreground propagation in video. In: ECCV (2014) 3
17. Jain, S.D., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: CVPR (2017) 3
18. Jang, W.D., Kim, C.S.: Semi-supervised video object segmentation using multiple random walkers,. In: BMVC (2016) 3
19. Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: CVPR (2017) 3
20. Jang, W.D., Kim, C.S.: Interactive image segmentation via backpropagating refinement scheme. In: CVPR (2019) 3
21. Jang, W.D., Lee, C., Kim, C.S.: Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In: CVPR (2016) 3

22. Koh, Y.J., Jang, W.D., Kim, C.S.: POD: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In: CVPR (2016) 3
23. Koh, Y.J., Kim, C.S.: Primary object segmentation in videos based on region augmentation and reduction. In: CVPR (2017) 3
24. Koh, Y.J., Lee, Y.Y., Kim, C.S.: Sequential clique optimization for video object segmentation. In: ECCV (2018) 3
25. Lempitsky, V.S., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV (2009) 3
26. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. IEEE Trans. Pattern Anal. Mach. Intell. **41**(6), 1515–1530 (2018) 3
27. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: From extreme points to object segmentation. In: CVPR (2018) 3
28. Najafi, M., Kulharia, V., Ajanthan, T., Torr, P.: Similarity learning for dense label transfer. In: CVPRW (2018) 10, 11
29. Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018) 3, 5
30. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: CVPR (2019) 4, 10, 11, 12
31. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019) 3
32. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV (2013) 3
33. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR (2017) 3
34. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016) 12
35. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 DAVIS challenge on video object segmentation. arXiv:1704.00675 (2017) 2, 3, 8, 11
36. Price, B.L., Morse, B.S., Cohen, S.: LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: ICCV (2009) 3
37. Ren, H., Yang, Y., Liu, X.: Robust multiple object mask propagation with efficient object tracking. In: CVPRW (2019) 10, 11
38. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: Interactive foreground extraction using iterated graph cuts. ACM Trans. on Graphics **23**(3), 309–314 (2004) 3
39. Shankar Nagaraja, N., Schmidt, F.R., Brox, T.: Video segmentation with just a few strokes. In: ICCV (2015) 3
40. Song, G., Myeong, H., Lee, K.M.: SeedNet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In: CVPR (2018) 3
41. Song, H., Wang, W., Zhao, S., Shen, J., Lam, K.M.: Pyramid dilated deeper convlstm for video salient object detection. In: ECCV (2018) 3
42. Tang, M., Gorelick, L., Veksler, O., Boykov, Y.: GrabCut in one cut. In: ICCV (2013) 3
43. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: CVPR (2017) 3
44. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: FEELVOS: Fast end-to-end embedding learning for video object segmentation. In: CVPR (2019) 3, 8, 13, 14

45. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017) 3
46. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Interactive video cutout. ACM Trans. on Graphics **24**(3), 585–594 (2005) 3
47. Wang, W., Shen, J., Porikli, F.: Saliency-aware geodesic video object segmentation. In: CVPR (2015) 3
48. Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S.C., Ling, H.: Learning unsupervised video object segmentation through visual attention. In: CVPR (2019) 3
49. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018) 6
50. Wu, J., Zhao, Y., Zhu, J.Y., Luo, S., Tu, Z.: MILCut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In: CVPR (2014) 3
51. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015) 8
52. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: CVPR (2016) 3
53. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: YouTube-VOS: A large-scale video object segmentation benchmark. arXiv:1809.03327 (2018) 3, 8
54. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR (2018) 3