WaterNet: An adaptive matching pipeline for segmenting water with volatile appearance

Yongqing Liang¹, Navid Jafari², Xing Luo³, Qin Chen⁴, Yanpeng Cao³, and Xin Li¹ (✉)

© The Author(s) 2020.

Abstract We develop a novel network to segment water with significant appearance variation in videos. Unlike existing state-of-the-art video segmentation approaches that use a pre-trained feature recognition network and several previous frames to guide segmentation, we accommodate the object's appearance variation by considering features observed from the current frame. When dealing with segmentation of objects such as water, whose appearance is non-uniform and changing dynamically, our pipeline can produce more reliable and accurate segmentation results than existing algorithms.

Keywords video segmentation; water segmentation; appearance adaptation

1 Introduction

1.1 Problem and approach

Semi-supervised video object segmentation (VOS) determines pixel-wise masks for objects of interest in a video sequence, starting from a given segmentation of the first frame given. This is an important task in video processing for such tasks as object identification, object tracking, and video editing. Recent deep-learning based VOS algorithms work well for segmenting everyday objects in commonplace

video. However, the performance of VOS algorithm often decreases significantly when objects in the video have changing appearance caused by illumination changes, or motion or deformation. For example, water often has a volatile appearance. The color and texture of water can vary between consecutive frames due to specular reflections, ripples, waves, turbulence, sediment concentration, etc. Such rapidly changing appearance often leads to poor water segmentation in videos.

Water is not the only case: appearance variations are common in practice. Examples include buildings with glass windows, along with cars or other objects with shiny paint or reflective surfaces. In this work we focus on segmenting water from videos, as it is a typical and representative object with dynamically changing appearance. In particular we consider water present as lakes, canals, rivers, floods, and so on.

In the semi-supervised VOS task, an annotated segmentation of the first frame is provided as part of the input. Most recent VOS techniques apply image semantic segmentation modules (e.g., fully convolutional networks (FCN) [1]) to learn the appearance of the object of interest. To tackle the appearance disparity between the training and test data, recent semi-supervised VOS algorithms usually adopt one of two architectures detection-based schemes, such as Refs. [2–11], compute and then propagate the segmentation of the past few frames to the current frame. Many approaches in this category require an online training process that adaptively fine-tunes the pretrained network to the object's specific appearance in the test video. Matching-based schemes [12–15] formulate video object segmentation as pixel-wise classification in a learnt embedding space. Such methods achieve promising results without online training.

1 School of Electrical Engineering and Computer Science, Louisiana State University, USA. E-mail: Y. Liang, yqliang@cct.lsu.edu; X. Li, xinli@cct.lsu.edu (✉).

2 Department of Civil Engineering, Louisiana State University, USA. E-mail: njafari@lsu.edu.

3 Department of Mechanical Engineering, Zhejiang University, China. E-mail: X. Luo, luoxing@zju.edu.cn; Y. Cao, caoy@zju.edu.cn.

4 Department of Civil Engineering, Northeastern University, USA. E-mail: q.chen@northeastern.edu.

Manuscript received: 2020-01-16; accepted: 2020-01-25

However, these methods are built upon the assumption that appearance does not change significantly in consecutive frames. If this assumption does not hold and the object in the current frame looks different from previous frames, such approaches become unreliable. In this work, we aim to develop a more reliable VOS pipeline for water (and other objects with changing appearance) in such more challenging scenarios.

We observe that features of water learnt from previous frames may change significantly and may not work well in identifying water pixels in the current frame. Figure 1 illustrates two example frames in one of our testing videos. Between two consecutive frames (a) and (b), the water's appearance (color, ripples, and also certain reflections) clearly changes, and the texture from previous frames can not effectively guide segmentation of the later frame. Indeed, in such scenarios, it is likely that water in the first frame may also look different and not provide good guidance. In Fig. 1(c), we draw a heatmap to show the l_2 -norm distance between the feature maps extracted from these two frames: the corresponding water regions

are quite different.

Our *main idea* is based on this aforementioned observation: the appearance of water (or other specular objects) may change dynamically and be difficult to predict, but its spatial locations and shapes in two consecutive frames are often more predictable and stable. Therefore, certain sub-regions identified in the previous frame, under some appropriately estimated transformations (e.g., obtained by simple tracking), are likely to be still occupied by the object in the current frame. These regions in the current frame provide valuable clues in learning the new appearance of this object. For example, in Figs. 1(d)–1(f), if we take water regions in the center (of the water region detected in the last frame), e.g., the green pixel region, as our reference, and use their feature vectors as templates, then other water regions in the current frame have better similarity to one of these reference regions.

1.2 Water segmentation dataset and benchmark

Another challenge in developing effective VOS systems is the lack of pixel-wise annotated training

Fig. 1 Appearance differences between frames. (a, b) Two consecutive frames, f_{28} and f_{29} , of a video from which we wish to segment the water region. Using our feature encoder trained on *WaterDataset*, the feature maps of f_{28} and f_{29} are very different; their l_2 -norm distance are visualized in (c). If a pixel in f_{29} , the green pixel in (d), is picked as a reference, features extracted from other water regions in f_{29} share better similarity with this reference. (d) color-encodes the l_2 -norm distance between the green pixel's feature vector and features of other regions. (e) The l_2 -norm distance when 5 reference pixels are selected. (f) The l_2 -norm distance when 20 reference pixels are selected. Green pixel regions are selected as references. When appearance changes dramatically, the spatial correlations of features may be stronger than their temporal correlations.

datasets. Specifically, for this water segmentation task, on the one hand, water-related image annotations are rather few, and on the other hand, water's appearance can be very varied. These make the learning of water appearance significantly difficult. For this work, we have thus built a water-related image database, which we referred to as the *WaterDataset*. This training dataset contains 2388 water-related images that come with annotations. It also contains 20 manually labeled water videos for testing. Our model and the comparative methods are all trained and evaluated using this dataset. This *WaterDataset* and the performance scores are available for use in future comparisons.

1.3 Contributions

The main contributions of this work are:

- a novel video object segmentation network for water, named *WaterNet*, which can effectively capture variations in water's appearance in video through online learning and updating, and
- a water segmentation database and benchmark to support image and video water segmentation research.

Our experiments demonstrate that our new pipeline clearly outperforms existing state-of-the-art VOS approaches in identifying water undergoing large variations in appearance. Our benchmark and source codes and the water segmentation dataset are available at <https://github.com/xmlyqing00/WaterNet>.

2 Related work

Video object segmentation (VOS) has been an active research topic for the last decade. Existing approaches can be generally classified as detection-based methods and matching-based methods.

2.1 Detection-based methods

Methods in this category segment objects from videos frame by frame. The pipelines of OSVOS [2], OSVOS-S [9], and OnAVOS [5] are similar to that of an FCN. Their models are trained on offline datasets. Given a test video with first frame annotation, they apply data augmentation to the first frame and use that to fine-tune their models. However, without temporal information, these methods may produce jittering segmentations because of object motions or appearance variations.

Recent approaches such as LucidTracker [4] and MSK [3] build neural networks that take the first frame annotation and masks of previous frames as inputs to create the mask for the current frame. Given a test video, most of these approaches heavily rely on online learning to remember the object appearance in this specific video. While these methods achieve strong performance, they require online training to recognize the target object, which takes an extra 10–20 minutes. RGMP [16] takes the first frame and the previous frame as references to predict object masks without online training. However, as shown in Fig. 1, if object appearance changes between frames, previous frames may not be able to effectively guide the segmentation and these methods can fail.

2.2 Matching-based methods

While there is a strong interest in semi-supervised video object segmentation by leveraging online training on the first frame annotation to achieve better performance, other approaches aim to obtain better runtime and performance without online training. Recent matching-based methods such as PML [13], VideoMatch [14], FAVOS [8], and FEELVOS [15] formulate the segmentation problem as a pixel-wise assignment task. These algorithms learn pixel-wise embedding spaces and maintain a set of feature templates to explicitly memorize the appearance of the target object in the reference image. At test time, a matching mechanism is designed to match the features of the current frame per pixel. These approaches update the feature templates after the segmentation of the each frame. However, when the appearance of the object changes suddenly between consecutive frames, feature templates built upon previous frames may not adapt to changes in the current frame: the outdated templates may not match features of the objects. In this work, we specifically design WaterNet to adapt to volatile appearance.

3 WaterNet segmentation

We now explain the design of our WaterNet, an appearance-adaptive network.

3.1 Overview

Given a sequence of N video frames $\{f_0, f_1, \dots, f_{N-1}\}$, and an annotation s_0 of the first frame in the

form of a mask indicating the object segmentation, we wish to compute the segmentation mask of the object in the subsequent video frames, denoted as $\{s_1, s_2, \dots, s_{N-1}\}$. The frames $f \in R^{H \times W \times 3}$ are in RGB space. The segmentation masks $s \in [0, 1]^{H \times W}$ are maps in which 0 indicates background and 1 indicates water.

Figure 2 illustrates the main pipeline of our proposed WaterNet. It consists of two branches: a parent network (*ParentNet*) and an appearance-adaptive branch (*AA-branch*). They share the same feature encoder E , which generates a feature map from an input image. The *ParentNet*, which is based on standard image semantic segmentation, is trained to learn the appearance of water from static images, and it predicts a binary water mask h_P for a given image frame f_t .

The *AA-branch* makes the segmentation adaptive to water appearance in the current video, which may look different from the training dataset and change from frame to frame. The *AA-branch* maintains three template sets: *initial-reference templates* T_I , *recent-frame templates* T_R , and *current-frame templates* T_C . Each template set is a list of feature vectors. The

feature encoder E extracts pixel features from the first frame, a few previous frames, and the current frame, respectively, and rearranges them into these three template sets. The feature map x_t of the current frame f_t is also extracted by E . The *similarity calculator* (SC) matches x_t with these three template sets to produce three water segmentations h_I , h_R , and h_C . They are fused to compose the *AA-branch* segmentation h_A .

Finally, the *ParentNet* segmentation h_P and the *AA-branch* segmentation h_A are combined to give the output segmentation s_t .

Note that in recent matching-based VOS algorithms [8, 13–15], features of the current frame are also compared with feature templates (obtained over a few previous frames) to estimate segmentation. However, because water has an inconsistent appearance, when its appearance changes suddenly between consecutive frames, features learnt from the past few frames cannot always effectively guide recognition in the current frame. The proposed *current-frame templates* in our *AA-branch* can use regions in the current frame as guidance to better accommodate such sudden appearance changes.

Fig. 2 Overview of WaterNet, which consists of a parent network (*ParentNet*) and an appearance-adaptive branch (*AA-branch*). They share the same feature encoder E , which generates features of input image. In *ParentNet* (blue background), a feature decoder D uses the current frame's feature x_t to predict a water segmentation h_P . In the *AA-branch* (yellow background), a deterministic similarity calculator matches features of the current frame x_t with feature templates T_I , T_R , T_C , to predict water segmentations h_I , h_R , h_C . Fusion modules merge these segmentations of the current frame to give the final segmentation s_t .

We now consider the components of our system in detail.

3.2 Parent Network

Our parent network *ParentNet* is based on an FCN and has two components: a feature encoder E and a feature decoder D , as shown in Fig. 3. The encoder E encodes appearance information from RGB space to the embedding space. We use $f_t \in R^{H \times W \times 3}$ to represent a frame in RGB space, and $x_t \in R^{h \times w \times c}$ to denote a feature tensor in the embedding space, where t is the time index, H and W are the width and height of the frame, h and w are the width and height of the feature tensor, and c is the number of feature channels. The ratio $W : w$ (and $H : h$) depends on the downsampling layers of E . The decoder D consists of a set of deconvolutional layers, which take the feature tensor x_t and also the features in the corresponding stream in E through skip connections, and generates a parent segmentation h_P . We build E based on ResNet-34 [17], with the last fully connected layers removed. Its weights are initialized from the ImageNet pre-trained model. After end-to-end training of ParentNet on the WaterDataset, E and D are used to generate feature tensors and parent segmentations, respectively.

3.3 Appearance-adaptive branch

The ParentNet learns the appearance of water from the offline static image data. But the appearance of water varies from video to video, and even frame to frame. Recent VOS approaches use the first frame annotation to fine-tune the parent network to enable it to recognize the appearance of water in this specific

Fig. 3 Overview of ParentNet. ParentNet consists of a feature encoder E and a decoder D . The encoder uses the ResNet-34 architecture and is pre-trained on ImageNet and then fine-tuned on *WaterDataset*.

video. However, information from the first frame still may not accurately reflect the water's current appearance in later frames. Also, online training often requires 10–20 minutes on modern GPU cards to retrain the network, which restricts the system's applicability to certain real-time flood monitoring and prediction tasks, for example.

Our appearance-adaptive branch (AA-branch) aims to tackle frame-to-frame appearance changes and provide better runtime efficiency. The AA-branch predicts a water mask h_A , which is later fused with the segmentation from the ParentNet to give the output segmentation. The pipeline of the AA-branch may be summarized as follows:

1. Initialize T_I and T_R using the annotation s_0 of the first frame f_0 , and its extracted feature map x_0 (see Section 3.3.1);
- For each subsequent time step $t \geq 1$:
2. Use E to extract the feature map x_t for f_t and get a parent segmentation h_P ; then create the current frame templates T_C by adding a subset of features from x_t (see Section 3.3.1);
3. Compare each region of f_t with T_I, T_R , and T_C (see Section 3.3.2), and then output the AA-branch segmentation h_A ;
4. Fuse h_A and h_P to give the final segmentation s_t for f_t (see Section 3.3.3);
5. Update as necessary (see Section 3.3.3).

3.3.1 Feature template settings

We maintain three active template sets to remember the water's appearance recently observed in the video. There are two types of templates: object (water) templates, and background templates, which are separated using a *feature splitter* module. Figure 4 shows the pipeline of the splitter module. The *feature splitter* FS reorganizes the feature maps generated by encoder E into a list of object templates $U_o \in R^{L_o \times c}$ and a list of background templates $U_b \in R^{L_b \times c}$ according to the given template mask MSK . The template mask MSK is defined as a binary image in which 0 represents the background and 1 represents the object. Therefore,

$$\begin{cases} U_o = \{Y(i) | MSK(i) = 1\} \\ U_b = \{Y(i) | MSK(i) = 0\} \end{cases} \quad (1)$$

where $i \in [1, hw]$ enumerates all regions in the feature map Y .

Initial-reference templates. The initial-reference templates T_I remember the initial appearance of the

Fig. 4 Pipeline of feature splitter FS . Feature encoder E takes the input RGB image to produce a feature map. The splitter is a deterministic module that divides the feature map into object feature templates and background feature templates according to the segmentation mask.

water. We first use the encoder E from the ParentNet to convert f_0 to the feature tensor $x_0 \in R^{h \times w \times c}$. Using the feature splitter module FS , we use the first frame mask s_0 to divide the feature map x_0 into object and background templates. These object and background templates are together the initial-reference templates T_I .

Recent-frame templates. We maintain recent-frame templates T_R containing features from the previous M frames to track recent water appearance. Like the initial-reference templates, T_R consists of object templates T_R^o and background templates T_R^b . Since we propagate the water segmentation frame by frame, using the segmentation of the previous frame s_{t-1} , we update the recent-frame templates for the current frame segmentation. The mask s_{t-1} of the feature map x_{t-1} of the previous frame is used to separate it into an object map V_o and a background map V_b . To provide more robust feature templates, we append new features from V_o and V_b to the recent-frame templates that both (i) have high segmentation scores (larger than a threshold T_{hc}), and (ii) are far from the object boundary (distance to the boundary larger than a threshold r_1). In addition, to restrict feature templates to a moderate size for computational efficiency, we remove features that were added M frames ago.

Current-frame templates. Unlike recent VOS approaches that only use previous frames to model object appearance features, we further model object appearance from reliable regions of the current frame. For example, in most of water videos we have observed, the water does not move significantly. Based on the segmentation s_{t-1} of the last frame,

the object's central region (its pixels that are far away from changing boundary) is almost always still occupied by the object in the current frame. More generally, if objects are moving, but their motion can be estimated by tracking or optical flow algorithms, then motion of the central region of the object could also be estimated. We denote such regions as *high-confidence regions*. We can then learn the object's up-to-date appearance from texture sampled in such high-confidence regions.

In our current implementation, high-confidence regions of the object and the background are extracted from the current frame f_t . E produces the feature map $x_t \in R^{h \times w \times c}$ of the current frame f_t . Mask s_{t-1} is a binary map where 0 represents background and 1 represents the water. In the high-confidence feature extractor HC module, let $U_o = s_{t-1}$ be the water mask and $U_b = \mathbf{1} - s_{t-1}$ be the background mask. We perform r_0 rounds of erosion operations on U_o and U_b to obtain high-confidence regions. Then, as for the feature splitter FS , we allocate the feature map of the high-confidence regions to the object template and the background template. These two templates form the current-frame templates T_c .

3.3.2 Feature matching

We compare the feature map x_t of the current frame f_t with the above three template sets T_I , T_R , and T_C to identify potential object regions. A *similarity calculator* (SC) provides efficient matching. It takes two inputs, the current frame features x_t and feature templates, and outputs a score map. Higher values in the score map indicate these regions have higher likelihood to be water. Figure 5 shows the details of the similarity calculator module.

Specifically, the object feature templates and background feature templates are initialized to U_o and U_b . Let the size of the object feature templates be m_w and the size of the background feature templates be m_b . Two similarity calculators compute an object score map and a background score map for the given feature map $x_t \in R^{h \times w \times c}$. The object score map H_o gives the regions' likelihood to belong to the object, and the background score map H_b to belong to the background. Let the feature vector of pixel i in the feature map of the current frame f_t be $x_t(i)$, $i \in \{1, \dots, hw\}$, the feature vector of an entry j_1 in the object feature templates be $U_o(j_1)$, $j_1 \in \{1, \dots, L_o\}$,

Fig. 5 Similarity calculator SC module. For each frame f_t , the encoder E generates a feature tensor x_t . The feature vector corresponding to each region i in f_t is $x_t(i)$. We compute the cosine similarity between each feature vector of the $x_t(i)$ and the object/background features in the template list. The object/background score map is the average of the top K similarity scores. The fusion module Fuse2 fuses the object/background score maps to give a segmentation mask.

and the feature vector of an entry j_2 in the background feature templates be $U_b(j_2)$, $j_2 \in \{1, \dots, L_b\}$. First, we compute cosine similarity between the feature map and templates using

$$\begin{cases} CS_o(i, j_1) = \frac{x_t(i)U_o(j_1)}{\|x_t(i)\| \|U_o(j_1)\|} \\ CS_b(i, j_2) = \frac{x_t(i)U_b(j_2)}{\|x_t(i)\| \|U_b(j_2)\|} \end{cases} \quad (2)$$

$$j_1 \in \{1, \dots, L_o\}, \quad j_2 \in \{1, \dots, L_b\}$$

where $i \in \{1, \dots, hw\}$, $CS_o \in [-1, 1]^{hw \times L_o}$ is a cosine similarity matrix between the feature map and the object templates, and $CS_b \in [-1, 1]^{hw \times L_b}$ is a cosine similarity matrix between the feature map and the background templates. L_o and L_b are the sizes of the feature templates.

We compute the object score H_o and background score H_b of the feature map x_t from the top K cosine similarity matrices CS_o and CS_b :

$$\begin{cases} H_o(i) = \frac{1}{K} \sum_{j=1}^K \text{top}_K(CS_o(i), j) \\ H_b(i) = \frac{1}{K} \sum_{j=1}^K \text{top}_K(CS_b(i), j) \end{cases} \quad (3)$$

where $i \in \{1, \dots, hw\}$, and $\text{top}_K(CS(i), j)$ is a function that returns the j -th largest similarity scores from the i -th row. K is set to 10 in our experiments.

Here H_o and H_b are the object and background segmentation matrices, with dimension $h \times w$. We upsample H_o and H_b to the original image size $H \times W$ using bilinear interpolation. Finally, we linearly combine H_o and H_b using the Fuse2 module to obtain the similarity-based segmentation seg :

$$seg = (H_o - H_b + 2)/4 \quad (4)$$

where the segmentation $seg \in [0, 1]^{H \times W}$ is the output of the similarity calculator module.

In the AA-branch, we deploy three similarity calculator modules and match the current frame feature x_t with the three feature templates T_I , T_R , and T_C , to obtain three object segmentations: the initial-reference-based segmentation h_I , the recent-frame-based segmentation h_R , and the current-frame-based segmentation h_C .

3.3.3 Segmentation fusion

The above three segmentations are fused, using a module named Fuse0, to give the current frame's appearance-adaptive segmentation h_A :

$$h_A = \lambda_0 h_F + \lambda_1 h_T + \lambda_2 h_C \quad (5)$$

where $\lambda_0 + \lambda_1 + \lambda_2 = 1$. We initialize $\lambda_0 = 0.4$, $\lambda_1 = 0.2$, $\lambda_2 = 0.4$ and gradually decrease λ_0 every 10 frames since the appearance of the first frame becomes less informative as the time goes on using:

$$\begin{cases} \lambda_0 = 0.9\lambda_0 \\ \lambda_1 = 1 - \lambda_0 - \lambda_2 \end{cases} \quad (6)$$

The weight for the current-frame segmentation remains unchanged.

We fuse the appearance adaptive segmentation h_A and the ParentNet segmentation h_P using another module, Fuse1, to obtain the final segmentation for the current frame f_t :

$$s_t = \lambda_A \cdot h_A + (1 - \lambda_A) \cdot h_P \quad (7)$$

where λ_A is a balancing factor.

3.4 Implementation details

Note that the initial-reference templates are constant during the evaluation, while the current-frame templates are updated for each frame. The recent-frame templates track features in the previous M frames.

WaterNet can be trained on a still water image dataset and evaluated on dynamic water videos. Once the ParentNet has been trained, the AA-branch can directly reuse the encoder E and decoder D from the ParentNet to extract feature maps. We use ResNet-34 [17] as the backbone of the encoder E . We set the

total epochs to 200, the initial learning rate to 0.1, and gradually decrease it during training. To train the ParentNet, we randomly pick an image and its ground-truth from the WaterDataset, and augment the training data following Ref. [4] by randomly adjusting colors and applying affine, flipping, and cropping transformations. During testing, we set $K = 10$, $M = 2$, $r_0 = 12$, $r_1 = 8$, $T_{hc} = 0.7$, and $\lambda_A = 0.5$, and run the whole WaterNet to predict the water mask.

4 Experiments

We have compared our proposed WaterNet with several state-of-the-art video object segmentation methods on our new benchmark, *WaterDataset*.

4.1 Dataset and evaluation metrics

Our new benchmark for the water segmentation task, named *WaterDataset*, includes a training set and an evaluation set. The training set has 2388 water-related still images with annotations; 1888 images are from ADE20K [18] and 300 images are from RiverDataset [19]. These images contain various types of water, including lakes, canals, rivers, oceans, and floods. The evaluation set contains 20 water related videos:

1. 7 videos recorded on days with heavy rain, when local creeks and ponds were flooded. Frames in these 7 videos were all manually labeled.
2. 10 surveillance videos from Farson Digital Watercams [20] that recorded open waters from 8 a.m. to 6 p.m. Frames in these 10 videos were uniformly labeled every 50 frames.
3. 3 surveillance videos taken at the beach that recorded changes in sea waves.

We adopt the evaluation measures used by the DAVIS Challenge [21, 22]. In particular we use region ($\mathcal{J}$) and boundary ($\mathcal{F}$) measures to evaluate segmentation quality. The region measure, also called Jaccard index, is a widely used evaluation metric in video object segmentation. It calculates the intersection-over-union (IoU) of the estimated mask and the ground-truth mask. We compute the mean IoU across all frames in the test videos. The boundary measure evaluates the accuracy of boundaries, via bipartite matching between the boundary pixels of both masks. Finally, $\mathcal{J}\&\mathcal{F}$ is the average of $\mathcal{J}$ and $\mathcal{F}$.

In addition, we adopt the three error measure statistics from Ref. [23]. Let $O = \{F_i\}$ be the dataset of video sequences and C be an error measure, either the region ($\mathcal{J}$) or boundary ($\mathcal{F}$) measure. First, the *mean* is the average error defined as

$$M_c = \frac{1}{|O|} \frac{1}{|F_i|} \sum_{F_i \in O} \sum_{f_j \in F_i} C(f_j) \quad (8)$$

Second, the *recall* measures the fraction of sequences scoring higher than a threshold τ , defined as

$$R_c = \frac{1}{|O|} \frac{1}{|F_i|} \sum_{F_i \in O} \sum_{f_j \in F_i} \mathbf{I}[C(f_j) > \tau] \quad (9)$$

where $\tau = 0.5$ and $\mathbf{I}$ is the indicator function having the value 1 when the condition is satisfied and the value 0 otherwise. Third, the *decay* measures how the performance changes over time. Let $Q_i = \{Q_i^1, Q_i^2, Q_i^3, Q_i^4\}$ be the partition of the video F_i into quartiles. We define

$$D_c = \frac{1}{|O|} \sum_{F_i \in O} |C(Q_i^1) - C(Q_i^4)| \quad (10)$$

For the mean and the recall measures, higher numbers are better, while for the decay measure, lower numbers are better.

4.2 Quantitative comparison

We compared our method with several state-of-the-art methods on the WaterDataset. Recent VOS approaches can be generally classified into three categories.

1. Detection-based methods such as OSVOS [2], OSVOS-S [9], and OnAVOS [5], which segment the video frame-by-frame without considering temporal consistency. We chose OSVOS as the representative approach of this category.
2. Propagation-based methods such as LucidTracker [4], MSK [3], and RGMP [16], which use the segmentation of the previous frame(s) to predict an object mask for the current frame. We chose RGMP as representative of this category because it outperforms other mask propagation methods.
3. Methods without online training such as PML [13], VideoMatch [14], FAVOS [8], and FEELVOS [15]. We chose FEELVOS [15] as the representative approach in this category, as it significantly outperforms PML [13] and VideoMatch [14].

All our experiments were performed on an Intel Xeon(R) E5-2630 v2 (2.60 GHz $\times$ 24) with a GTX 1080Ti GPU card and 32 GB RAM.

Table 1 documents the comparison of WaterNet with these state-of-the-art methods. We use the

Table 1 Comparison of WaterNet with other state-of-the-art methods on *WaterDataset*. Region ($\mathcal{J}$) calculates the intersection-over-union (IoU) of the estimated mask and the ground-truth mask. Boundary ($\mathcal{F}$) evaluates the accuracy of boundaries. Mean is the average error. Recall measures the fraction of sequences scoring higher than a threshold. Decay measures how performance changes overtime. Mean and Recall are the two most important measures

	$\mathcal{J}\&\mathcal{F}$ -Mean	$\mathcal{J}$ -Mean	$\mathcal{J}$ -Recall	$\mathcal{J}$ -Decay	$\mathcal{F}$ -Mean	$\mathcal{F}$ -Recall	$\mathcal{F}$ -Decay
OSVOS ⁻	0.382	0.559	0.643	0.008	0.207	0.000	0.002
OSVOS	0.597	0.726	0.902	0.113	0.467	0.420	0.137
RGMP	0.491	0.647	0.780	0.013	0.337	0.193	0.011
FEELVOS	0.569	0.681	0.713	0.069	0.457	0.461	0.027
WaterNet (ours)	0.645	0.822	0.937	0.070	0.468	0.452	0.087

super-script “⁻” to denote a method for which online training was disabled. As for OSVOS [2], we followed the authors’ pipelines to fine-tune the model with the first frame annotation. Note that OSVOS requires an extra 10 minutes for segmenting each video. OSVOS achieves 0.597 for $\mathcal{J}\&\mathcal{F}$ -Mean compared with 0.382 from OSVOS⁻. Online training does improve segmentation accuracy, but we can see that OSVOS has the worst decay scores, as segmentation performance decreases over time. We conclude that online training cannot adapt to appearance changes during the video.

In terms of region measure ($\mathcal{J}$), WaterNet outperforms the other methods, as its three feature templates help capture the changing appearance of water. In terms of boundary measure ($\mathcal{F}$), WaterNet’s $\mathcal{F}$ -Recall is little weaker than FEELVOS’s, as FEELVOS adopts a strong neighbor filter that only considers features in a small window, which improves the boundary measure, but may fail if the object moves dramatically. Note that the *decay* measures how the segmentation results change over time. Because OSVOS⁻ is an image-based method and it ignores temporal information, it achieves good

decay scores while its segmentation results are poor: only 0.382 for $\mathcal{J}\&\mathcal{F}$ -Mean. In terms of overall measure $\mathcal{J}\&\mathcal{F}$ -Mean, WaterNet achieves the highest score 0.645 of the methods compared.

4.3 Qualitative evaluation and comparisons

4.3.1 Appearance difference between the first frame and the test frame.

Figures 6 and 7 visualize segmentation results for the tested methods on the test videos “Buffalo0” and “Stream3”. “Buffalo0” is a time-lapse video taken near Houston’s Buffalo Bayou during Hurricane Harvey in August 2017. The bayou was flooded, and our goal is to track the water elevation at this location during that time. “Stream3” is a video taken near a local creek on campus during heavy rain in August 2018.

In Fig. 6, the first frame was captured at 07:55 while the test frame was captured at 13:25. Different solar altitudes make the water look distinct. In Fig. 7, different weather conditions (wind and rain) make the appearance of the water dissimilar. Online training based methods (such as OSVOS) and first frame guided methods (such as RGMP) fail in this case because the appearance of the test frame is

Fig. 6 Qualitative results from test video “buffalo0”: (a) 1st frame, (b) 12th frame for segmentation, (c) ground-truth water, (d)–(h) segmentations of WaterNet, RGMP, OSVOS without online training, OSVOS, and FEELVOS.

Fig. 7 Qualitative results from test video “stream3”: (a) 1st frame, (b) 11th frame for segmentation, (c) ground-truth water, (d)–(h) segmentations of WaterNet, RGMP, OSVOS without online training, OSVOS, and FEELVOS. Red box highlights artifacts in (e) and (h).

very different from the first frame’s. Our model outperforms other methods as track appearance changes during evaluation.

4.3.2 Appearance difference between two consecutive frames

Figure 8 shows segmentation results for the tested methods on the test video “Boston Harbor”, taken near Boston Harbor in February 2019. From the 8th frame to the 9th frame, although the camera position is fixed, the appearance of the water is quickly affected by reflections, shadows, and waves. Figure 9 shows results for the test video “Holiday Inn Beach”. The appearance of the sea is highly dynamic in the video. Mask propagation based methods such as RGMP and FEELVOS fail in this case because they exploit the information of the previous frames to segment the current frame. Such a mechanism works poorly when object appearance in consecutive frames changes greatly. Our model has an appearance-adaptive branch, which captures the appearance of the object by the high-confidence features observed in the current frame. The segmentation results show

that our model is more robust to appearance variation in such scenarios as well.

4.4 Ablation study

We also analyzed the effectiveness of the key components of our model, through two variants. One was to remove the module which matches current frame features and current-frame templates (see Section 3.3.1). The other one was to remove the entire appearance-adaptive branch to assess the performance of the ParentNet (see Section 3.3).

4.4.1 WaterNet without current-frame templates

When processing each frame, WaterNet compares current frame features with the current-frame templates to identify water regions. We set the weight of current-frame segmentation $\lambda_2 = 0$ and tested our model without current-frame templates. Without this procedure, our model’s $\mathcal{J}\&\mathcal{F}$ decreases from 0.645 to 0.638. WaterNet without current-frame templates still performs better than matching-based approaches such as FEELVOS, mainly for two reasons: (i) our module weights in the AA-branch are adaptive,

Table 2 Ablation study of two variations of our model: (1) WaterNet without current-frame templates. (2) WaterNet without AA-branch. Region ($\mathcal{J}$) is IoU of the output mask and the ground-truth mask. Boundary ($\mathcal{F}$) evaluates accuracy of the boundaries. Mean is the average error. Recall measures the fraction of sequences scoring higher than a threshold. Decay measures how performance changes over time. Mean and Recall are the two most important measures

	$\mathcal{J}\&\mathcal{F}$ -Mean	$\mathcal{J}$ -Mean	$\mathcal{J}$ -Recall	$\mathcal{J}$ -Decay	$\mathcal{F}$ -Mean	$\mathcal{F}$ -Recall	$\mathcal{F}$ -Decay
WaterNet w/o AA-branch	0.479	0.603	0.706	0.032	0.355	0.188	0.013
WaterNet w/o current-frame	0.638	0.812	0.917	0.063	0.465	0.484	0.080
WaterNet	0.645	0.822	0.937	0.070	0.468	0.452	0.087

Fig. 8 Qualitative results from test video “Boston Harbor”: (a) 8th frame, (b) 9th frame for segmentation, (c) ground-truth water, (d)–(h) segmentations of WaterNet, RGMP, OSVOS without online training, OSVOS, and FEELVOS.

Fig. 9 Qualitative results from test video “Holiday Inn Beach”: (a) 8th frame, (b) 9th frame for segmentation, (c) ground-truth water, (d)–(h) segmentations of WaterNet, RGMP, OSVOS without online training, OSVOS, and FEELVOS.

and we decrease the weight of the initial-reference templates and increase the weight of the recent-frame templates as time goes on, and (ii) our recent-frame templates track features from the past M frames, while FEELVOS only utilizes features from the last frame.

4.4.2 WaterNet without AA-branch

Our WaterNet consists of two components: ParentNet and the AA-branch. The appearance-adaptive branch

maintains a set of feature templates to identify the object in each frame. We removed the AA-branch and ran our model on ParentNet. Because ParentNet is an image-based segmentation network which does not consider temporal information, although the resulting performance is more stable, it is less accurate. Note that mean and recall are the two most important measures. Without the AA-branch, the score of the $J\&F$ -Mean decreases from 0.645 to 0.479.

5 Conclusions

5.1 Summary

We developed an adaptive matching pipeline, WaterNet, to tackle appearance change in water in video object segmentation. Our main idea is to use the object's appearance as observed in the current frame to help its identification and segmentation. We built an annotated dataset of water images and videos, to facilitate water-related image and video segmentation. Our experiments demonstrated that with our new AA-branch, the accuracy of VOS on appearance-changing objects clearly improves, and our WaterNet outperforms existing state-of-the-art algorithms in video water segmentation.

5.2 Limitations

The feature templates are updated based on each frame's segmentation result without supervision. If in some frame the segmentation is incorrect, the derived feature templates and the estimated high-confidence region could also be incorrect, which would negatively impact further segmentation accuracy. This is also a problem in existing approaches where segmentations of the past few frames are used to guide the subsequent segmentation. We will study the relationship between appearance change and other information and priors such as saliency, attention, or tracking information, and explore the possibility of integrating these priors and preprocessing mechanisms to help tackle this issue.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant EAR 1760582, and the Louisiana Board of Regents ITRS LEQSF(2018-21)-RD-B-03. We would like to express our appreciation to anonymous reviewers whose comments helped improve and clarify this manuscript.

References

- [1] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.
- [2] Caelles, S.; Maninis, K. K.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; Van Gool, L. One-shot video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 221–230, 2017.
- [3] Perazzi, F.; Khoreva, A.; Benenson, R.; Schiele, B.; Sorkine-Hornung, A. Learning video object segmentation from static images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2663–2672, 2017.
- [4] Khoreva, A.; Benenson, R.; Ilg, E.; Brox, T.; Schiele, B. Lucid data dreaming for multiple object tracking. *arXiv preprint arXiv:1703.09554*, 2017.
- [5] Voigtlaender, P.; Leibe, B. Online adaptation of convolutional neural networks for video object segmentation. In: Proceedings of the British Machine Vision Conference, 116.1–116.13, 2017.
- [6] Hu, Y.-T.; Huang, J.-B.; Schwing, A. G. MaskRNN: Instance level video object segmentation. In: Proceedings of the 31st Conference on Neural Information Processing Systems, 325–334, 2017.
- [7] Luiten, J.; Voigtlaender, P.; Leibe, B. PRMIVOS: Proposal-generation, refinement and merging for video object segmentation. In: *Computer Vision – ACCV 2018. Lecture Notes in Computer Science, Vol. 11364*. Jawahar, C.; Li, H.; Mori, G.; Schindler, K. Eds. Springer Cham, 565–580, 2018.
- [8] Cheng, J. C.; Tsai, Y. H.; Hung, W. C.; Wang, S. J.; Yang, M. H. Fast and accurate online video object segmentation via tracking parts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7415–7424, 2018.
- [9] 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 Transactions on Pattern Analysis and Machine Intelligence* Vol. 41, No. 6, 1515–1530, 2019.
- [10] Yang, L. J.; Wang, Y. R.; Xiong, X. H.; Yang, J. C.; Katsaggelos, A. K. Efficient video object segmentation via network modulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6499–6507, 2018.
- [11] Li, X. X.; Loy, C. C. Video object segmentation with joint re-identification and attention-aware mask propagation. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11207*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 93–110, 2018.
- [12] Yoon, J. S.; Rameau, F.; Kim, J.; Lee, S.; Shin, S.; Kweon, I. S. Pixel-level matching for video object segmentation using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2186–2195, 2017.

- [13] Chen, Y. H.; Pont-Tuset, J.; Montes, A.; Van Gool, L. Blazingly fast video object segmentation with pixel-wise metric learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1189–1198, 2018.
- [14] Hu, Y. T.; Huang, J. B.; Schwing, A. G. VideoMatch: Matching based video object segmentation. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11212*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 56–73, 2018.
- [15] Voigtlaender, P.; Chai, Y. N.; Schroff, F.; Adam, H.; Leibe, B.; Chen, L. C. FEELVOS: Fast end-to-end embedding learning for video object segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9473–9482, 2019.
- [16] Oh, S. W.; Lee, J. Y.; Sunkavalli, K.; Kim, S. J. Fast video object segmentation by reference-guided mask propagation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7376–7385, 2018.
- [17] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778, 2016.
- [18] Zhou, B. L.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ADE20K dataset. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5122–5130, 2017.
- [19] Lopez-Fuentes, L.; Rossi, C.; Skinnemoen, H. River segmentation for flood monitoring. In: *Proceedings of the IEEE International Conference on Big Data*, 3746–3749, 2017.
- [20] Farson digital watercams. <https://www.farsondigitalwatercams.com/>. Accessed: 2019-09-30.
- [21] Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.
- [22] Caelles, S.; Pont-Tuset, J.; Perazzi, F.; Montes, A.; Maninis, K. K.; Van Gool, L. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019.
- [23] 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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 724–732, 2016.

Yongqing Liang received his B.S. degree in computer science from Fudan University, China, in 2017. He is currently a Ph.D. student in the School of Electrical Engineering and Computer Science, Louisiana State University, USA. His research interests include visual data understanding, computer vision, and computer graphics. He uses tools for the analysis of massive volumetric images. He specialises in high performance computing on clusters and GPUs.

Navid Jafari received his B.S. degree in civil engineering from the University of Memphis in 2010. He received his M.S. and Ph.D. degrees in 2011 and 2015, respectively, from the University of Illinois at Urbana-Champaign in the Department of Civil & Environmental Engineering. He is currently an assistant professor at Louisiana State University in the Department of Civil & Environmental Engineering, where his research is focused at the intersection of geotechnical and coastal engineering with natural hazards. He is specifically focused on the performance of natural infrastructure, natural and man-made slopes, and flood protection infrastructure during hurricanes.

Xing Luo majored in mechanical engineering, receiving his B.E. degree from the University of Science and Technology Beijing in 2018. He is currently pursuing a Ph.D. degree in the Institute of Manufacturing Technology and Automation, Zhejiang University. His research interests include multi-modal image processing and analysis.

Qin Chen is a professor of Civil & Environmental Engineering and Marine & Environmental Sciences at Northeastern University. He specializes in the development and application of numerical models for coastal dynamics, including ocean waves, storm surges, nearshore circulation, fluid-vegetation interaction, and sediment transport and morphodynamics. His research includes field experiments and application of remote sensing and high-performance computing technologies to solve engineering problems. He leads the Coastal Resilience Collaboratory funded by the NSF CyberSEES award.

Yanpeng Cao is a research fellow in the School of Mechanical Engineering, Zhejiang University, China. He graduated with M.Sc. degree in control engineering (2005) and Ph.D. degree in computer vision (2008), both from the University of Manchester, UK. He worked in a number of R&D institutes

such as Institute for Infocomm Research (Singapore), Mtech Imaging Ptd Ltd (Singapore), and National University of Ireland Maynooth (Ireland). His major research interests include infrared imaging, sensor fusion, image processing, and 3D reconstruction.

Xin Li received his B.S. degree in computer science from the University of Science and Technology of China in 2003, and his M.S. and Ph.D. degrees in computer science from Stony Brook University (SUNY) in 2008. He is currently an associate professor with the School of Electrical Engineering and

Computer Science and the Center for Computation and Technology, Louisiana State University, USA. He leads

the Geometric and Visual Computing Laboratory at LSU. His research interests include geometric and visual data processing and analysis, computer graphics, and computer vision.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.