Autonomous Robots
https://doi.org/10.1007/s10514-020-09950-9

Enhancing the morphological segmentation of microscopic fossils through Localized Topology-Aware Edge Detection

Qian Ge[1] · Turner Richmond[1] · Boxuan Zhong[1] · Thomas M. Marchitto[2,3] · Edgar J. Lobaton[1]

Received: 13 January 2020 / Accepted: 29 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
Fossil single-celled marine organisms known as foraminifera are widely used in oceanographic research. The identification of species is one of the most common tasks when analyzing ocean samples. One of the primary criteria for species identification is their morphology. Automatic segmentation of images of foraminifera would aid on the identification task as well as on other morphological studies. We pose this problem as an edge detection task for which capturing the correct topological structure is essential. Due to the presence of soft edges and even unclosed segments, state-of-the-art techniques have problems capturing the correct edge structure. Standard pixel-based loss functions are also sensitive to small deformations and shifts of the edges penalizing location more heavily than actual structure. Hence, we propose a homology-based detector of local structural difference between two edge maps with a tolerable deformation. This detector is employed as a new criterion for the training and design of data-driven approaches that focus on enhancing these structural differences. Our approaches demonstrate significant improvement on morphological segmentation of foraminifera when considering region-based and topology-based metrics. Human ranking of the quality of the results by marine researchers also supports these findings.

Keywords Edge detection · Topological structure · Morphological segmentation

1 Introduction

Foraminifera, also known as forams, are ubiquitous ocean dwelling amoeboid organisms whose shells (typically less than 1 mm in diameter) are widely used in oceanographic and geoscience research. They are common in many modern and ancient marine environments, and as such have become invaluable tools for petroleum exploration (Tipsword 1962), paleoecology (Berggren 1992), biostratigraphy (Kennett and Srinivasan 1983), paleobiogeography (Berggren 1972), and paleoclimatology (Rohling and Cooke 1999). One of the most common tasks associated with forams is the identification of species from ocean sediment or rock samples. As different species live in different environments and at different geologic times, fossil foram species found in samples are used for determining environmental or climate conditions in the past and the relative ages of sediment layers. A sample for study can contain thousands of forams, and the current practice is to have students and scientists identify the species manually. This is a tedious, time consuming and error prone process. Therefore, an automatic visual foram species identification system is desirable. Since foram shell characteristics, such as aperture location and chamber shape and arrangement (see Fig. 1a for some examples), are the primary criteria for species identification (Kennett and Srinivasan 1983), segmentation of chambers and apertures from images of forams would provide powerful features for automatic species identification. Foram segmentation can also

This work was supported by US National Science Foundation Grants OCE-1637023, OCE-1637039, OCE-1829970 and OCE-1829930.

This is one of the several papers published in Autonomous Robots comprising the Special Issue on Topological Methods in Robotics.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-020-09950-9) contains supplementary material, which is available to authorized users.

✉ Edgar J. Lobaton
edgar.lobaton@ncsu.edu

[1] Department of Electrical and Computer Engineering, North Carolina State University, Box 7911, Raleigh, NC 27695, USA

[2] Institute of Arctic and Alpine Research, University of Colorado, Boulder, CO 80309, USA

[3] Department of Geological Sciences, University of Colorado, Boulder, CO 80309, USA

Published online: 11 November 2020

assist in morphological studies of foramnifera shells (Corliss 1991; Boltovskoy et al. 1991) by eliminating the need for manual selection and measurement. We make use of edge detection as the mechanisms for achieving morphological segmentation.

Edge detection is a crucial low-level operation in computer vision and it is commonly used as a pre-processing step for image segmentation (Chen et al. 2015; Kirillov et al. 2016), object detection (Ferrari et al. 2008; Zitnick and Dollar 2014) and object tracking (Zhu et al. 2015; Choi and Christensen 2012). Data driven edge detection approaches were first proposed by Dollár et al. (2006) using a Probabilistic Boosting Tree to classify edge/non-edge pixels by extracting edge features from a small neighborhood around each pixel. Since then, learning-based edge detection has been extensively studied and its performance has improved through better features (Lim et al. 2013; Iandola et al. 2014), more sophisticated training strategies (Liu and Lew 2016), more advanced classifiers (Dollár and Zitnick 2015) and more powerful network deep neural network architectures (Xie and Tu 2017). In spite of the great success in pixel level performance, these strategies are not suitable in applications such as edge-based image segmentation where local topology and connectivity information is essential. Standard pixel-based metrics greatly penalize small shifts/deformations in edge maps and do not differentiate them from topological error in the edge maps (e.g., gaps or extra edges) that are essential for morphological characterization of our specimen. Hence, there is a need to develop ways to detect and localize topological error in a segmentation and develop mechanisms for enforcing deep neural networks to correct them.

In this article, we introduce a new methodology for image segmentation, which makes use of localized topological criteria to enhance data-driven approaches to edge detection, and apply it to morphological segmentation of forams. The scientific contributions in this paper include: (1) the development of a Localized Topology-Aware (LTA) detector of errors in segmentation results which is invariant to small deformations; (2) the integration of the detector in the training and design of data-driven approaches for edge detection; and (3) the validation of our approach on an existing foram dataset (Zhong et al. 2017; Ge et al. 2017). The LTA error detection makes use of invariances captured by localized relative homology conditions. Two strategies are considered for the integration of this approach with data-driven techniques, as shown in Fig. 1. The first strategy (TA-Loss) biases a pixel-based loss function to focus on the patches which violate our topological conditions. The second approach (TA-Correct) proposed a network structure that predicts the detection of topological errors and uses a generative model to fix those locations. Besides demonstrating the improvement of our approach with these strategies using standard metrics, we also evaluate the performance of our approach by having an expert and two novices evaluate the quality of each segmentation and rank the results based on their overall quality. The proposed methodologies show significant improvement over the baseline approach when considering region-based and topology-based metrics. Human rankings of our results also supports these findings.

The remainder of this paper is organized as follows: Sect. 2 gives an overview of the related work; Sect. 3 introduces our novel Localized Topology-Aware (LTA) Error Detection approach and two Topology-Aware edge detection strategies (TA-Loss and TA-Correct); experiments as well as the analysis of the results are discussed in Sect. 4; and Sect. 5 summarizes our contributions and the future work.

2 Related work

2.1 Data-driven edge detection

Lots of studies have been carried out the past 50 years in edge detection, contour prediction and edge-based segmentation. Specifically, with the recent success of deep neural networks (DNN) in computer vision, numerous DNN-based methods have been proposed (Bertasius et al. 2015; Liu et al. 2019; Xie and Tu 2017; Xu et al. 2017; Zhu et al. 2018), significantly pushing the field forward. HED (Xie and Tu 2017), for instance, achieves the state-of-the-art image-to-image edge detection performance via a novel holistically-nested architecture. Liu et al. (2019) fully exploit and utilize the multiscale and multilevel features from pre-trained convolutional networks to further improve the performance.

However, most methods still rely on pixel-level losses, failing to encode high-level geometry and topology information. Several studies have explored incorporating geometric errors into the loss (BenTaieb and Hamarneh 2016; Rojas-Moraleda et al. 2017; Qu et al. 2018; Mosinska et al. 2018). For example, StripNet (Qu et al. 2018) segments long and continuous strip patterns by first considering the segmentation as a boundary-regression problem, then applying the topological constraints on the predicted boundaries. Also, Mosinska et al. (2018) propose a topology-aware loss to implicitly impose the topological constraints on edges by minimizing the distance of high-level features between the prediction and the ground truth labeling.

2.2 Class imbalance in dense prediction

Prediction tasks such as edge detection and object detection always face class imbalance during training as only a small part of training samples are positive. Class-balanced cross-entropy loss function proposed in Xie and Tu (2017) and class-balanced sampling discussed in Bansal et al. (2017) are commonly used in edge detection tasks. For object detec-

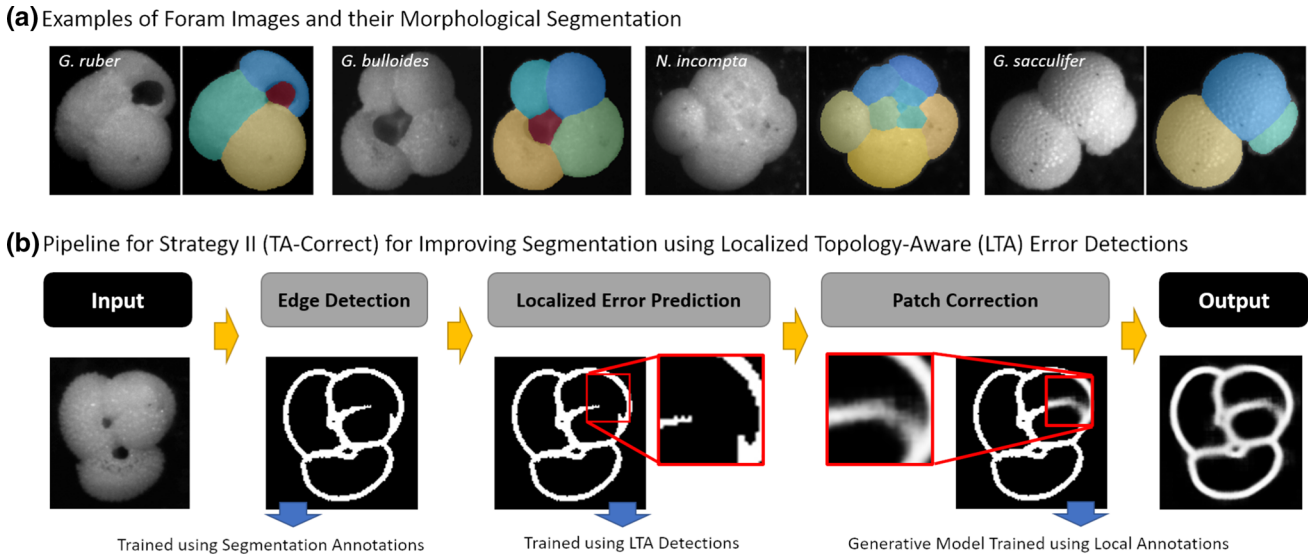


Fig. 1 Overview of the proposed pipeline. **a** Examples of foram images of different species and their corresponding morphological segmentation. The red region corresponds to the aperture, and the other regions are the chambers. We pose this as an edge detection problem for which we compare a baseline approach against approaches that focus on local topological errors in the output. Our first strategy (TA-Loss) for improving segmentation biases a traditional pixel-based loss to focus on patches with topological errors. **b** Our second strategy (TA-Correct) trains a network to detect local errors in a segmentation by using our Localized Topology-Aware (LTA) detection as ground truth, and corrects the errors using a generative model over the small patch (color figure online)

tion tasks, hard example mining is used to select the hardest negative samples for class-balanced training (Shrivastava et al. 2016; Girshick 2015). Besides class-balanced sampling approaches, focal loss which is proposed in Lin et al. (2017) adds a factor to the cross-entropy loss to make the training focus more on hard misclassified samples. Our TA-Loss strategy is similar to class-balanced cross-entropy loss and focal loss in weighting the loss for different pixels. However, the proposed approach puts higher weights on pixels causing the topology difference, thus the model focuses on preserving the topological structure created by the edges.

2.3 Homology-based approaches

In Letscher and Fritts (2007), edge directed topology is used to define regions of an image from the output of an edge detection algorithm. Regions with similar features are merged, in ascending order of the disc size needed to create the region (i.e. α-complex), to produce the final segmentation. In Beksi and Papanikolopoulos (2016), a region growing approach is proposed for 3D region segmentation through persistent homology analysis of point cloud data. In Hu et al. (2019), a continuous-valued loss function is proposed to preserve the topological structure of a segmentation by forcing both, the result segmentation and the ground truth, to have the same Betti number. In Clough et al. (2019), a topological pixel-wise gradient is computed by identifying pixels which affect the persistence of desired topological features to incorporate topological prior knowledge into the MRI images segmentation. Our methodology for LTA Error Detection in the segmentation builds on a framework developed for occlusion detection (Lobaton et al. 2010) and image feature matching (Lobaton et al. 2011) using topological invariants. A localized homology-based metric was also proposed in Ahmed et al. (2014) for comparison of roadmap reconstructions, which has a methodology similar to our approach. However, this metric does not accommodate for local deformations.

3 Methodology

As discussed in Sect. 1, a Topology-Aware edge detection network can be trained by paying close attention to locations causing topological errors in the segmentation. We accomplish this by proposing the Localized Topology Aware (LTA) Error Detector (Sect. 3.1) and by incorporating this technique into two strategies (Sects. 3.2, 3.3) for enhancing the edge detection.

The need for the LTA Error Detector is illustrated in Fig. 2. As previously discussed, data-driven approaches for edge segmentation often rely on pixel-based losses. As shown in Fig. 2b, in which two edge maps are compared, these losses greatly penalize small shifts/deformations in edge maps as shown in the left plot, and do not differentiate them from errors (such as gaps) in the edge maps that are more essential for segmentation. In this case, the LTA Error Detection (rightmost plot in Fig. 2b) shows the desired type of detection that

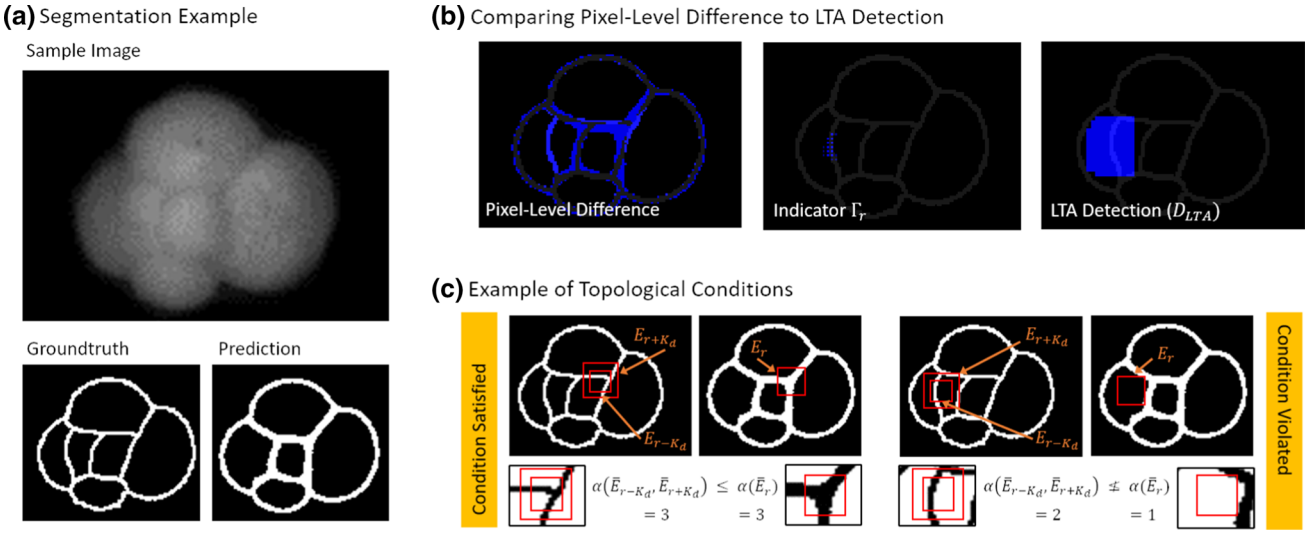


Fig. 2 Localized Topology-Aware (LTA) Error Detection applied to Foram segmentation. **a** Illustration of the segmentation output. Groundtruth labels (bottom-left) and sample segmentation output (bottom-right) are shown. **b** Comparing pixel-based difference and topological detection for sample output in **a**. Metrics such as cross-entry, are based on image difference (left) which can highlight areas were edges map are just misaligned. The proposed LTA detection (right) ignores small deformations and highlights locations where the local topological structure does not match. **c** Illustration of topological conditions. On the (left), neighborhoods in which the localized topological conditions on the local complement sets $\bar{E}_r$ are satisfied even though a small perturbation is present. On the (right), a case in which the conditions are not satisfied due to a missing edge

would enable data-driven approaches to automatically focus on these errors.

3.1 Localized Topology-Aware (LTA) Error Detection

Tools from topological data analysis (TDA) (Edelsbrunner et al. 2000) including persistence diagram (Zomorodian and Carlsson 2005) and persistence landscape (Bubenik 2015) have previously been used as topological features of a set of points in a space. Both of these approaches provide robust global topological descriptors by summarizing the filtration, a sequence of topological features at different scales, of point clouds. However, for dense prediction applications such as edge detection, local descriptors are more desired. Although the persistence diagram and persistence landscape can be locally applied to a small neighborhood around each pixel in the image, high time complexity makes utilizing these methods prohibitively expensive. As such, we propose a novel topology-based detector to locally indicate the difference between two sets of edges which is invariant to bounded deformations. This methodology builds on a framework developed for occlusion detection (Lobaton et al. 2010) and image feature matching (Lobaton et al. 2011) using topological invariants.

First, we define the edge map functions $I_k : \Omega_k \to \{0, 1\}$, where $\Omega_k \subset \mathbb{R}^2$ and a value of 1 indicates the presence of an edge. We consider $k \in \{1, 2\}$ when comparing a ground truth edge map against a generated edge map. The order of the edge maps is not relevant since the detector will make use of symmetric topological conditions between both maps.

Definition 1 Two maps **match** if

$$I_2(x) = I_1 \circ g(x) \tag{1}$$

for some deformation $g : \Omega_2 \to \Omega_1$ with $||g(x) - g(x')|| \leq K_d||x - x'||$, where K_d is a Lipschitz deformation bound.

Our objective is to identify any locations in which two edge maps fail to match. In this case, we interpret failing to match as not being able to find any deformations that locally match the structure of the edge maps. Our approach will verify a number of conditions based on local topological invariants that should be satisfied if a local deformation exists, and flag as violating locations those that do not satisfy these invariants.

In order to define the topological invariants, let us begin by defining $E_k = \{x \in \Omega_k \mid I(x) = 1\}$ as the edge set, $B_r(x) = \left\{x' \in \mathbb{R}^2 \mid ||x - x'||^2 < r^2\right\}$ as the ball of radius r centered at x, $E_{k,r}(x) = E_k \cap B_r(x)$ as the local edge set, and its local complement as $\bar{E}_{k,r}(x) = E_k^c \cap B_r(x)$. We can check that if two edge maps match given a deformation g then

$$E_{1,r-K_d}(x) \subset g\left(E_{2,r}(x)\right) \subset E_{1,r+K_d}(x). \tag{2}$$

This follows from $B_{r-K_d}(x) \subset g\left(B(x)\right) \subset B_{r+K_d}(x)$ which is a consequence of having K_d as the Lipschitz bound of

g. Let us further define the count $\alpha(S)$ of connected components over a set S, and the relative count $\alpha(S_-, S_+)$ for $S_- \subset S_+$. The relative count corresponds to the number of connected components in S_- after those that belong to the same connected components in S_+ are identified. This is essential computing relative homology. Then, we have that:

Lemma 1 *Given two edge maps, I_1 and I_2, that match under a local deformation bounded by K_d then for any $r > K_d$ the following condition is satisfied:*

$$\alpha\left(E_{1,r-K_d}(x), E_{1,r+K_d}(x)\right) \leq \alpha\left(E_{2,r}(x)\right). \tag{3}$$

Proof Let $L(S)$ be the set of connected components over a set S, and $L(S_-, S_+)$ be the set of connected components in S_- after they have been identified using the connected components in S_+. We let $S := E_{2,r}(x)$, $S_- := E_{1,r-K_d}(x)$ and $S_+ := E_{1,r+K_d}(x)$. In order to prove the desired result, we just need to define an injective mapping $i : L(S_-, S_+) \to L(S)$. Let $l \in L(S_-, S_+)$, then we know that l was formed from connected components in S_-; hence by Eq. 2, we have that these sets are all included in $g(S)$. This gives a candidate set of connected components in S that can be associated with l. We assigned any of them to l as part of our definition of i. Note that any component in $g(S)$ is included in a single component in S_+ (by Eq. 2). Since the list of components in $g(S)$ that were candidates for l as part of our definition of i were selected from components in S_- that are included in a single component in S_+, then these candidates are not candidates for an element $l' \in L(S_-, S_+)$ for which $l' \neq l$ because they are associated with a different component in S_+. This shows that our mapping is one-to-one and concludes the proof. □

Note that this condition does not rely on g anymore, and it is purely a topology constraint over a local neighborhood. Building on this condition and definition below, we can have the result in Theorem 1.

Definition 2 Given two edge maps I_1 and I_2, we define an indicator map $\Gamma_r(x|I_1, I_2, K_d)$ at scale r and bound K_d as a binary map that has value 1 at location x if the following conditions are not satisfied

$$\begin{aligned}
&\alpha\left(E_{1,r-K_d}(x), E_{1,r+K_d}(x)\right) \leq \alpha\left(E_{2,r}(x)\right)\\
&\alpha\left(E_{2,r-K_d}(x), E_{2,r+K_d}(x)\right) \leq \alpha\left(E_{1,r}(x)\right)\\
&\alpha\left(\bar{E}_{1,r-K_d}(x), \bar{E}_{1,r+K_d}(x)\right) \leq \alpha\left(\bar{E}_{2,r}(x)\right)\\
&\alpha\left(\bar{E}_{2,r-K_d}(x), \bar{E}_{2,r+K_d}(x)\right) \leq \alpha\left(\bar{E}_{1,r}(x)\right)
\end{aligned} \tag{4}$$

Theorem 1 *Two edge maps I_1 and I_2 cannot be matched under a local deformation bounded by K_d if the indicator map $\Gamma_r(x|I_1, I_2, K_d)$ is not an all zero map.*

Proof If there exists a deformation g with Lipschitz deformation bound as defined in Eq. 1, then the first condition in Eq. 4 would need to be satisfied by the previous lemma. Since the deformation is also a deformation for the map $\bar{I}_k = 1 - I_k$, then the third condition must also be true. Finally, since g^{-1} is also a deformation between $\Omega_1 \to \Omega_2$ with the same Lipschitz constant, the second and fourth conditions must also hold. If any of these conditions are violated at any point, then such deformation does not exist, and as such, the edge maps cannot be matched. □

Figure 2c illustrates how the topological conditions are satisfied for small deformation and not for topological violations together with the corresponding indicator mask in Fig. 2b. The indicator Γ_r helps identify locations in which the neighborhood $B_r(x)$ could not match between the two edge maps. Hence, we define our LTA Error Detection as

$$D_{LTA}(I_1, I_2; K_d) = \{x \mid \Gamma_r(x|I_1, I_2, K_d) = 1\} \oplus B_r, \tag{5}$$

where $A \oplus B_r = \cup_{a \in A} B_r(a)$ is the dilation of set A by B_r. We will make use of the area of this mask as a measure of how well the two masks match. Alternatively, we could also compute the indicator over a range of K_d, $[K_{min}, K_{max}]$, and then assign to each location x a value corresponding to the smallest K' such that $\Gamma_r(x|I_1, I_2, K) = 0$ for all $K > K'$ in the range. If no value of K' is found then it means that no match can be found over the range of deformations. If a value of K' is found then it means that any smaller value of K has no deformation which makes the edge maps topologically equivalent. This new function can also be used to specify other performance metrics for comparing edge maps. Figure 6 shows some of the LTA detection when comparing manual and estimated segmentation.

As an implementation note, thick edge maps may include single pixel or a few pixels within the edges that are marked as non-edges, which would be counted on the lower bounds associated with $\bar{E}_{k,r}$ in Eq. 4. These artifacts are mainly due to the application of a threshold on the edge probabilities output by the prediction network in order to get an edge map, as such we do not consider them as actual errors. In order to remove these artifacts, we erode by a single-pixel the sets $\bar{E}_{1,r-K_d}(x)$ and $\bar{E}_{2,r-K_d}(x)$. Similarly, in order to handle narrow edges which (after sampling) can lead to disconnected edges maps with segments that may be one pixel apart from each other, we dilate the sets $E_{1,r+K_d}(x)$ and $E_{2,r+K_d}(x)$. These modifications do not affect the validity of the topological conditions, and provide a more robust detection by directly handling sampling issues.

3.2 Strategy I: Topology-Aware Edge Loss (TA-Loss)

In this section, we describe how the LTA detection can be used to bias a pixel-based loss in order to further refine patches with topological errors. For simplicity, we consider

the loss for a single pair of edge maps (groundtruth I_g and prediction I_p). We model the prediction as an edge detection network of the form $I_p(x) = f_W(x) \in [0, 1]$, where W represents the set of network parameters. We use all the edge pixels as positive samples and uniformly sample the same number of non-edge pixels as negative samples. The pixel-level loss is defined as

$$C_0(x; W) = -I_g(x) \cdot \log(I_p(x)) - (1 - I_g(x)) \cdot \log(1 - I_p(x)). \quad (6)$$

We also tried the class-balanced cross entropy loss and found equally sampling provided better performance. The LTA error detection between I_g and $I_p > \tau$ for some threshold τ does not directly specify which pixels cause the topological conditions to be violated, but instead specifies a neighborhood containing these points. Hence, we use it as as a criterion of hard example mining during the training phase. That is, we put more weights on pixels with non-zero values in the LTA detection. Hence, the *Topology-Aware Loss* (TA-Loss) is defined as

$$C_{TA}(x; W) = \begin{cases} \gamma C_0(x; W) & \text{if } x \in D_{LTA} \\ C_0(x; W) & \text{otherwise,} \end{cases} \quad (7)$$

where $\gamma \geq 1$ is a hyper parameter to adjust the influence of the topological metric. When $\gamma = 1$, the TA-Loss is equivalent to the original cross entropy loss. The TA-Loss strategy corresponds to using the loss above for refining an edge-detection network.

3.3 Strategy II: TA Edge Correction (TA-Correct)

In Sect. 3.2, the LTA detections are used for re-weighting the cross-entropy loss during training to force the network to pay more attention to errors causing topological differences between the predicted edge map and groundtruth. This approach is able to provide better performance on patch and topology metrics as shown in the experiment section. However, the training speed is around 4 times slower than the original cross entropy loss due to the computation of the LTA detections for each edge map at every training step. However, given a predicted edge map, we want to only refine the patches with edge gaps and keep the rest of patches the same. Thus, an attention mechanism can be used. That is, attention patches (patches containing detection errors) are first extracted, then fed into an edge generation network to be corrected and finally mapped back to the original edge map. In our early stage experiments, we tried to learn the attention patches in an unsupervised manner similar to Mnih et al. (2014). However, we found it is difficult to train a network to find the error patches without supervision. Since the LTA detection is able to indicate the patches with topological difference between two edge maps, we can make use of it to annotate those patches on predicted edge maps and then train an object detection network to learn to extract the patches potentially containing errors. Figure 1b illustrates the pipeline of this approach. First, patches with possible topological errors are identified (Sect. 3.3.1). Then each patch is re-scaled to a fixed size and is corrected (Sect. 3.3.2). Finally, the patches are mapped back to the original edge map and the entire edge map is smoothed by an edge smooth module.

3.3.1 Localized error prediction

In our experiments, an object detection network is used to predict patches containing topological errors on estimated binary edge maps. To create a training set for the gap patch detection, a set of estimated edge probability maps is generated by an edge detection network trained through the original cross entropy loss, and binary edge images are obtained by applying several different threshold values. Then, the LTA error detection are computed on the binary images to compare against the groundtruth images. Finally, bounding boxes are drawn to contain each connected component on the LTA detection as annotations to train the object detection network. To make sure the generative model could get enough clues for correcting the patch, the bounding boxes are enlarged to include some parts of edges as well. Figure 3a shows examples of labeled binary edge maps.

The object detection network YOLOv3 (Redmon and Farhadi 2018) was trained using this detection set to learn to extract patches with topological errors from the predicted edge maps. YOLOv3 is chosen in our experiment because it is fast and the accuracy of bounding boxes in our task is not as important as object detection tasks. Figure 3b shows the detection results on the test dataset. Finally, these bounding boxes are used to crop the binary edge maps, which are then re-scaled to a fixed size and used as input for the next step.

3.3.2 Patch correction using a generative model

In order to better match the standard notation of Conditional Variational Autoencoeders (CVAE) (Sohn et al. 2015), which is the type of generative model selected for patch correction, we define $\mathbf{x}$ as the input binary edge map (i.e., the output of the edge detection network) and $\mathbf{y}$ as the groundtruth edge annotation. Let us first motivate the need for such a generative model. A simple alternative for correcting patches is to train a fully convolutional neural network through the cross entropy loss. Through this approach, the probability of an edge pixel at location (i, j) (i.e., $\mathbf{y}(i, j) = 1$) given its neighborhood patch $N_{\mathbf{x}}(i, j)$ is modeled by a Bernoulli distribution conditional on $N_{\mathbf{x}}(i, j)$, which is equal to the conditional expectation of the random variable $\mathbf{y}(i, j)$. That is, the prob-

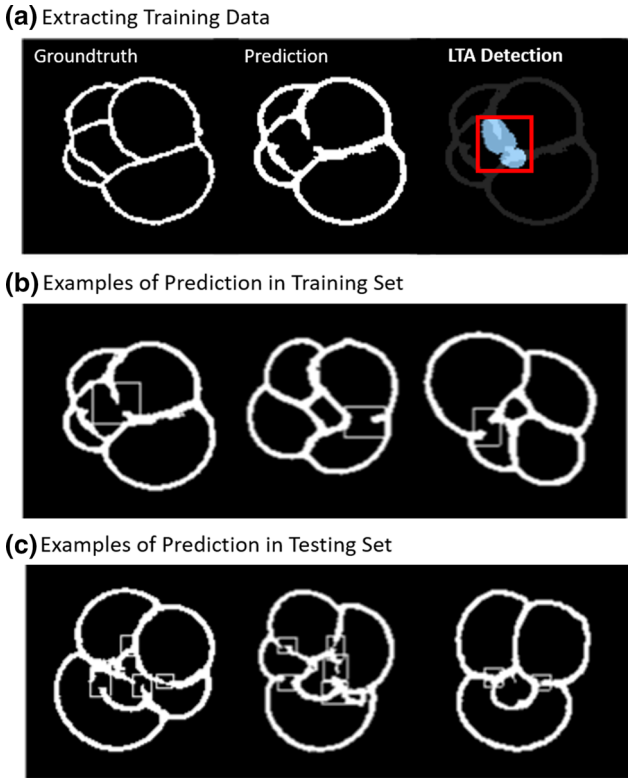


Fig. 3 Localized Topological Error Prediction. **a** Illustration of LTA detection used to extract bounding box used as an annotation for YOLOv3. **b** Examples of detection results on the training set. We can see that the first detection matches the training point shown in **a**. **c** Example of detection results on the testing set

ability we want to estimate is $p(\mathbf{y}(i,j) = 1|N_{\mathbf{x}}(i,j)) = \mathbb{E}\left[\mathbf{y}(i,j)|N_{\mathbf{x}}(i,j)\right]$. The expectation is empirically approximated by averaging over the training set. As there are may be multiple possible correct instances of the edge map (e.g., due to small shifts of the edge maps when soft edges are present), the predicted edges can be blurred by averaging these multiple possible outcomes. When attempting to recover the binary edges by applying a threshold on this likelihood, we could end up with missing edge sections due to the spreading of the likelihood. This blurring effect is an artifact that is also observed in other work (e.g., Walker et al. 2016). However, in our task, we only care about getting an appropriate instance of the gap filling. That is, during testing, we only need the model to predict one representative outcome. Then, a model which is able to perform probabilistic inference is more suitable for our task. Thus, a CVAE (Sohn et al. 2015) model is used.

Variational Autoencoders (VAE) (Kingma and Welling 2013) are generative models built from directed graphical models. In our case, each edge map $\mathbf{y}$ is assumed to be generated by a random process with parameters θ and latent variable $\mathbf{z}$. Thus the aim of the VAE is to learn two distributions: the prior distribution of the latent variables $p_\theta(\mathbf{z})$ and the conditional distribution $p_\theta(\mathbf{y}|\mathbf{z})$. If no simplifying assumptions about the marginal or posterior probabilities are made, this results in an intractable posterior $p_\theta(\mathbf{z}|\mathbf{y})$. Due to the intractability of the distribution, an approximation of the posterior is learned instead. The posterior approximation is known as the encoder and represented as $q_\phi(\mathbf{z}|\mathbf{y})$, which has parameters ϕ. Similarly, the conditional distribution, $p_\theta(\mathbf{y}|\mathbf{z})$, is known as the decoder.

Similar to the VAE, the Conditional Variational Autoencoder (CVAE) (Sohn et al. 2015) aims to learn the parameters for the posterior approximation of a conditional distribution, but includes a conditional input variable $\mathbf{x}$ such that the prior distribution of the latent variable is modeled by $p_\theta(\mathbf{z}|\mathbf{x})$. In this context, $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x}))$ provides a lower bound for $\log p_\theta(\mathbf{y}|\mathbf{x})$, which is used as the loss function for optimization of the CVAE. An empirical version of this bound obtained by sampling $z_l \sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ is given by

$$\mathcal{L}(\mathbf{x},\mathbf{y};\theta,\phi) = \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}_l) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})), \quad (8)$$

where L is the number of samples used to approximate the expectation.

In our task, the CVAE is applied on each detected patch, and the latent variable that generates the filled gap patch is conditional on the input detected patch. A first estimate $\hat{\mathbf{y}}_0$ of the corrected image is obtained using a CNN. The output patches of the CVAE are then re-scaled to the original resolution and added element-wise back to $\hat{\mathbf{y}}_0$ in order to obtain a new correction $\hat{\mathbf{y}}_1$. The final edge map $\hat{\mathbf{y}}$ is obtained by applying a single convolutional layer (prediction layer) to smooth the boundary of the patches. Figure 4 illustrates this process.

During training, the loss is computed on each detected patch as well as the entire edge map. Let N be the number of detected patches on one edge map, and $\mathbf{r}_x^n$ and $\mathbf{r}_y^n$ be the nth input patch and label patch. Then, we define our loss function as

$$\mathcal{L}_{CVAE}(\mathbf{x},\mathbf{y};\theta,\phi) = \frac{\nu_1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}_l) + \eta_1\sum_{n=1}^{N}\mathcal{L}_{Patch}\left(\mathbf{r}_x^n,\mathbf{r}_y^n;\theta,\phi\right), \quad (9)$$

where

$$\mathcal{L}_{Patch}\left(\mathbf{r}_x^n,\mathbf{r}_y^n;\theta,\phi\right) = \frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(\mathbf{r}_y^n|\mathbf{r}_x^n,\mathbf{z}_l^n\right) + - \mathrm{KL}\left(q_\phi\left(\mathbf{z}^n|\mathbf{r}_x^n,\mathbf{r}_y^n\right)||p_\theta\left(\mathbf{z}^n|\mathbf{r}_x^n\right)\right), \quad (10)$$

$\mathbf{z}_l$ is the set of $\mathbf{z}_l^n$, and η_1 and ν_1 are hyper-parameters to weight the importance of the patch loss and entire edge map loss. We choose $q_\phi(\mathbf{z}^i|\mathbf{r}_x^i, \mathbf{r}_y^i)$ and $p_\theta(\mathbf{z}^i|\mathbf{r}_x^i)$ to be Gaussian, and $p_\theta(\mathbf{r}_y^i|\mathbf{r}_x^i, \mathbf{z}_l^i)$ and $p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_l)$ to be Bernoulli distributions.

As discussed in Sohn et al. (2015), during training, both input and label images are used to draw sample $\mathbf{z}$ while only the input image is used during testing, which introduces a gap between training and testing pipeline. In order to narrow the gap, the network is also trained by setting $q_\phi(\mathbf{z}^n|\mathbf{r}_x^n, \mathbf{r}_y^n) = p_\theta(\mathbf{z}^n|\mathbf{r}_x^n)$ to make the training and testing pipelines be consistent. Then similar to Sohn et al. (2015), the corresponding Gaussian Stochastic Neural Network (GSNN) loss function can be written as

$$\mathcal{L}_{GSNN}(\mathbf{x}, \mathbf{y}; \theta, \phi) = \frac{1}{L}\sum_{l=1}^{L}\left(\eta_2 \sum_{n=1}^{N} \log p_\theta\left(\mathbf{r}_y^n|\mathbf{r}_x^n, \mathbf{z}_l^n\right) + \nu_2 \log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_l)\right), \tag{11}$$

where η_2 and ν_2 are hyper-parameters to weight the importance of the patch loss and entire edge map loss.

Thus, the total loss of the generative model for correcting the patches can be written as

$$\mathcal{L} = \rho\mathcal{L}_{CVAE} + (1-\rho)\mathcal{L}_{GSNN}, \tag{12}$$

where ρ is a hyper-parameter to balance the two losses.

During testing, the latent variable $\mathbf{z}^n$ for nth detected patch is drawn from the prior distribution $p_\theta(\mathbf{z}^n|\mathbf{r}_x^n)$ through the prior network. The output patch is generated from the conditional distribution $p_\theta(\mathbf{r}_y^n|\mathbf{r}_x^n, \mathbf{z}^n)$ through the generation network and added to the estimate $\mathbf{y}_0$ (output by a CNN). The final output edge map is then smoothed by the prediction layer. The training and testing process are summarized in Fig. 4. As suggested in Sohn et al. (2015), to improve the performance, the prior network also takes the initial guess $\hat{\mathbf{r}}_y$ obtained by the CNN and prediction layer as input. Also note that both prediction layers in Fig. 4 share parameters.

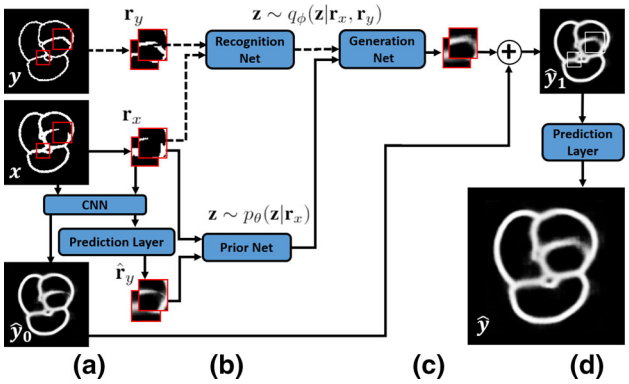


Fig. 4 Training and testing process of CVAE. Solid arrows are shared in both training and testing process, while dash arrows are only used during training. **a** Patches are detected on the input binary edge map $\mathbf{x}$ and also extracted from the label $\mathbf{y}$. The initial guess $\hat{\mathbf{r}}_y$ is obtained from $\mathbf{r}_x$ through the CNN and prediction layer. The initial estimate of the edge map $\hat{\mathbf{y}}_0$ (normalized for illustration purpose) is obtained from the input $\mathbf{x}$. **b** The latent variable $\mathbf{z}$ is drawn from the recognition network or prior network, depending on which path is used. **c** Output patches are generated through the generation network and added element-wisely to $\hat{\mathbf{y}}_0$ in order to get $\hat{\mathbf{y}}_1$. **d** The final prediction is smoothed by a prediction layer

4 Experiments

In this section, we first evaluate our LTA detection using other topology metrics proposed in Wegner et al. (2013) on a synthetic dataset for which we have control over the errors introduced. Then, we evaluate the impact of our Topology-Aware strategies on a real foram segmentation dataset.

4.1 Dataset

4.1.1 NCSU-CUB Foram (AROS 2020)

This dataset is used for visual foram species identification (Ge et al. 2017; Zhong et al. 2017). Images of each sample are taken under 16 different lighting directions via a microscope. It contains 514 manually segmented images which separate chambers and apertures of each sample. This dataset contains soft and hard edges, similar adjacent regions and non-closed edges, which makes edge detection non-trivial and is suitable for evaluating the performance of edge closing. In our experiment, the images are rescaled to 128×128 to reduce the computational complexity while keeping enough edge information. Among all the labeled samples, 322 samples are used for training and the remaining 192 samples are used for testing. Examples are shown in Fig. 1a.

4.1.2 Synthetic dataset

This dataset is inspired by the dataset above and it is designed for the study of iterative edge detection refinement. First, a 3D object with three to six intersecting ellipsoids (each of a different color) is randomly generated. Then, the ground truth image is created by taking a snapshot from a random viewpoint and extracting the edges of the ellipsoids. Due to the different colors of the ellipsoids, edges can be easily obtained by comparing the intensity of neighboring pixels. To mimic the imperfect predicted edge probability map, small gaps are randomly added to the inner edges, and each edge branch

is randomly thickened or thinned and assigned a probability value in a range of [0.2, 1.0]. The outer boundaries are not modified since they are easy to detect in the **NCSU-CUB Foram** dataset. Finally, images are rescaled to 128 × 128. This dataset is designed to directly quantify the power of the approaches to complete any missing local structure. The source code to create this dataset is available in AROS (2020). An example is shown in Fig. 7(left).

4.2 Analysis of LTA error detection

In this section, we first discuss how parameters r and K_d affect the LTA detection. Then, we evaluate our approach using the set of topology metrics from Wegner et al. (2013) to demonstrate its capability to quantify the topology difference between two edge maps.

4.2.1 Effect of LTA parameters

As discussed above, the parameters r and K_d used for the conditions in Eq. 4 are hyper-parameters of our approach. Figure 5a illustrates the average area of the LTA detection as a function of K_d when fixing r from 5 to 15. These curves were computed from the edge detection results of the **NCSU-CUB Foram** testing set obtained by the **Baseline** approach (see Sect. 4.3) with an edge threshold of 0.4, which provides the best performance. As K_d controls the level of deformation, more pixels get detected using smaller K_d. The area of the detection increases as r increases, which makes sense since larger neighborhoods are more likely to incorporate some topological error. In order to analyze how r and K_d affect the accuracy of the topological error detection, we also manually labelled the pixels causing topological difference between the predicted and ground truth edge map. We compute precision and recall plots (see Fig. 5b) of the LTA detection with respect to the labeled mask by fixing r from 6 to 12 and varying K_d from 0 to $r-1$ for each r. Some manually labeled samples are shown in Fig. 6d. To make the manually labels comparable to our LTA detection, the labeled regions are also dilated by the corresponding r as discussed in Sect. 3.1.

Since larger values of r involve more computation and less spatial accuracy, and there is a performance jump from $r = 8$ to $r = 9$ based on Fig. 5b, the parameter pair $r = 9$ and $K_d = 4$ could be an optimal choice when considering the trade off between speed and accuracy. However, in our experiments, we pick $r = 8$ and $K_d = 5$ for both the TA-Loss and YOLOv3 annotation. We observed that large r values, such as 9, tend to merge two detection regions creating large bounding boxes containing multiple topological errors which makes the patch correction step less likely to succeed since it focuses on larger more complex regions [see Fig. 6(bottom two lines)]. A K_d value of 4 (and smaller) detects some shifts in the edge map [see Fig. 6(top two lines)].

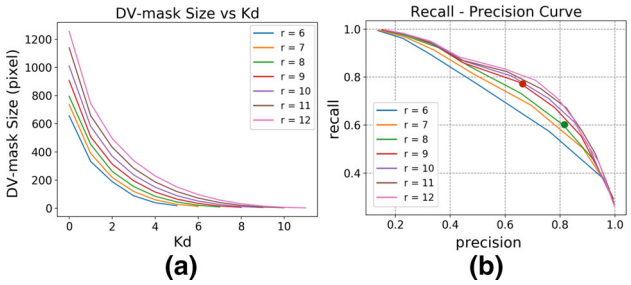


Fig. 5 **a** Average area of LTA detection for fixed r as a function of K_d in the foram dataset. **b** Precision–recall curve obtained by varying K_d and for fixed r when compared to manual labels of errors. Red and green points indicate $r = 9$ with $K_d = 4$ and $r = 8$ with $K_d = 5$, respectively (color figure online)

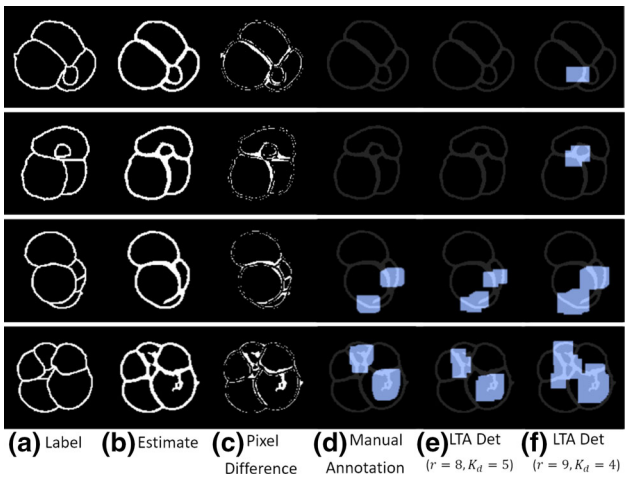


Fig. 6 Pixel difference and LTA detection of estimated edge maps obtained by the **Baseline** approach. Manual labels are dilated by $r = 8$. (Top two rows) D_{LTA} detection with $K_d = 4$ detects more regions caused by edge shift. (Bottom two rows) D_{LTA} detection with $r = 9$ merges two detected regions. More examples of D_{LTA} detection results with $r = 8$ and $K_d = 5$ can be found in the Supplement

4.2.2 LTA performance

In Fig. 6, we show pixel difference and LTA detection regions for the estimated edge maps obtained by the **Baseline** approach. A large part of the pixel differences result from small shifts and width differences of edges. However, the LTA detection results only highlight the regions causing structural differences. In order to demonstrate that the proposed methodology has the capacity to quantify the topology difference between two edge maps, we compare our results using a set of topological metrics proposed in Wegner et al. (2013). The metrics described in Wegner et al. (2013) involve the following three steps: (1) randomly sample two points lying on edges existing in both edge maps; (2) find the shortest path along the edge connecting the two points in each edge map; and (3) compare the length of the path between

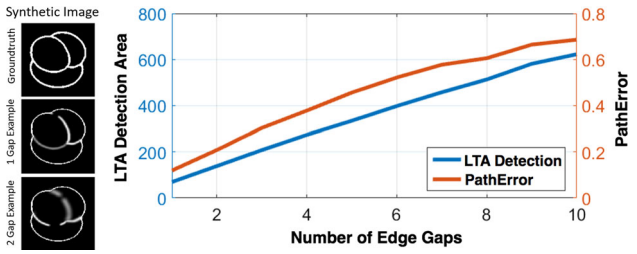


Fig. 7 Sample of synthetic image dataset (left) including groundtruth (top), edge probability maps with one gap (middle) and two gaps (bottom). Comparison of LTA detection area (left y-axis) and PathError (right y-axis) (right). Best viewed in color (color figure online)

the two edge maps. While sampling, we restrict our samples to those for which the two sampled points are always connected in the ground truth edge map. Then, a pair of points is labelled as *infeasible*, if the two points are disconnected in the predicted edge map. Additionally, if the difference of path length is larger than 10% of the path on ground truth, the pair is labelled as *2long2short*, otherwise it is labelled as *correct*. To summarize these three quantities into a single value, we define *PathError* as

$$PathError = \frac{2long2short + infeasible}{correct + 2long2short + infeasible}, \tag{13}$$

which quantifies the portion of sampled point pairs with different shortest paths in the ground truth and the predicted edge map.

To compare both types of metrics, we use a set of synthetic image pairs with controlled topology differences. Each pair of images consists of a ground truth image from the **Synthetic Image** dataset and a corresponding gap image created by randomly adding gaps on the blurred image of this dataset. The topology difference are controlled by the number of gaps in each image. That is, image pairs with more gaps have larger topology difference. We create ten sets of synthetic pairs, with the number of gaps ranging from 1 to 10—each set contains 200 image pairs. The results of the PathError metric and the area of the LTA detection averaged over 200 images on each set are reported in Fig. 7. We use $r = 6$ and $K_d = 2$ for this synthetic dataset. Both quantities consistently increase as the number of gaps increases as expected, which demonstrates that the proposed LTA detection is able to quantify the topology difference between edge maps as well as the *PathError* metric. Moreover, our local topology metric also provides the pixel locations causing the topology difference, which cannot be obtained from *PathError*.

4.3 Description of approaches and implementation details

In this section, we first discuss the network architecture of modules used in our edge detection experiments and then provide the implementation details. We compare six different approaches for edge detection. They include the Richer Convolutional Features **RCF** (Liu et al. 2019), **U-Net** (Ronneberger et al. 2015), **Linknet** (Chaurasia and Culurciello 2017), and three approaches of a modified D-LinkNet architecture. We review the D-LinkNet architectures more closely below. Each of **RCF**, **U-Net** and **Linknet** were modified slightly to accommodate the **NCSU-CUB Foram** dataset. The input to all of the models is a greyscale image. The models were retrained on the dataset for a single channel as most pretrained models used RGB and were trained on data not similar our dataset. Due to the small size of the images in the dataset, modifications are made for the architecture of **RCF** and the input image to **U-Net**. For **RCF**, the coarsest edge features from stage 5 were removed from the model. For **U-Net**, each image was padded with zeros to produce an image of 316×316 so that the output matched our original 128×128 size as closely as possible.

The next three models are all built around the D-LinkNet (Zhou et al. 2018) edge detection model, which is originally designed for road extraction. This model has an encoder-decoder architecture with skip connections and also a center dilation part in between. It has the advantage of extracting narrow and long span of regions in the image. The detailed architecture can be found in the Supplement. We use the same architecture as described in Zhou et al. (2018). However, we do not use the ImageNet (Deng et al. 2009) pre-trained model to initialize the encoder because our images are far different from natural images. Instead, we train the network from scratch.

In order to improve our edge detection performance, we adopt an iterative refinement approach (Mosinska et al. 2018). Instead of obtaining the predicted edge map through a single step, the output edge probability map is again used as the input of an additional edge detection model. This also makes the comparison between the approaches more fair since the **TA-Correct** approach is a refinement procedure by design. The first step of the **Baseline** and **TA-Loss** models consist of D-LinkNet models with the standard crossentropy loss (Eq. 6). The second step of the **Baseline** is also a D-LinkNet model with standard crossentropy, while for the **TA-Loss** we use the D-LinkNet with the weighted loss in Eq. 7 with $\gamma = 3$. We did not observe any additional improvement for $\gamma > 3$. For both approaches, the models are trained end-to-end by adding the loss of each step with weights of 1 and 3 for the first and second steps, respectively.

For the **TA-Correct** approach, each step is trained separately. The first step consists of a D-LinkNet model with

the standard crossentropy loss. The second step used the YOLOv3 and CVAE modules as described in Sect. 3.3. The patch losses are set to $\eta_1 = \eta_2 = 5$ and the entire edge map losses are set to $\nu_1 = \nu_2 = 1$ in Eqs. 9 and 11. The weight ρ in Eq. 12 is set to be 0.5. The object detection network YOLOv3 used in our experiments is implemented based on the original technique report (Redmon and Farhadi 2018), except that the bounding boxes are only predicted on two scales of feature maps and multi-scale training is not used, as we do not expect too large or too small gap regions. The CVAE is implemented based on the original paper (Sohn et al. 2015) to generate patches with filled edges. The CNN is implemented as an encoder-decoder structure, and the prediction layer is implemented as a single convolutional layer with a filter size 3×3. The detailed architecture can be found in the Supplement.

The scale to compute the LTA Error Detection for both the **TA-Loss** and YOLOv3 annotation (in the **TA-Correct** strategy) is set to be $r = 8$ and $K_d = 5$ as explained in Sect. 4.2.1. The local topological metric computation is implemented in C++ and other modules are implemented in Python using the TensorFlow framework (Abadi et al. 2015). The source code is available in AROS (2020). All the models are trained by Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is 2 for all models. The **Baseline**, **TA-Loss** and the first step of **TA-Correct** are trained end-to-end for 300 epochs with an initial learning rate of 0.001, which is then divided by 10 for every 100 epochs. The refinement step of **TA-Correct** are trained for 60 epochs with a learning rate 0.001.

4.4 Evaluation metrics

We use three types of metrics (pixel-based, region-based and topology-based) to quantitatively evaluate the edge detection results.

4.4.1 Pixel-based metrics

The standard *F1* score is sensitive to small shifts of the edges while the edges still maintain the topological structure. Therefore, we also use the *correctness* and *completeness* proposed in Wiedemann et al. (1998) as the relaxed precision and recall, and use *quality* (Wiedemann et al. 1998) to summarize them. The true positives are determined by the predicted and target edge pixels within a certain threshold distance, which makes it less sensitive to small shifts. In our experiments, we use a threshold distance of 5 pixels. See Wiedemann et al. (1998) for more details.

4.4.2 Region-based metrics

We use the region-level metrics to indicate the performance of the topological structure preservation. A binary edge map is first obtained by thresholding the edge probability map. Then, a closing operation using a disk with a radius of 2 pixels is used to refine the edge map. Finally the segmentation is obtained by extracting regions based on the skeleton of the edges. To evaluate the segmentation performance, the unweighted *Intersection over Union* (IoU) (Arbelaez et al. 2011) is adopted. We also use the *region recall* (Ge et al. 2017) to further evaluate how well the edge closure performs. This metric quantifies the percentage of the target regions detected in the predicted segmentation and a target region is marked as detected if there exists a predicted region with IoU greater than a threshold. In our experiment, the region recall thresholds 0.5 and 0.7 are used for evaluation.

4.4.3 Topology-based metrics

To directly evaluate the topology preserving performance, the *PathError* metrics (*correct*, *infeasible* and *2long2short*) introduced in Sect. 4.2.2 are used, and 300 paths are sampled per image for convergence of the evaluation results.

4.5 Performance on Foram images

Examples of the resulting predictions from the test set of the three approaches are shown in Fig. 8. Overall, the results of TA-Correct are closest to the ground truth labeling. It provides more closed edges as well as the clearest edge maps (see the windows w1, w2, w3 and w5 in Fig. 8). Compared with the **Baseline**, the TA-Loss has more closed edges and sometimes successfully detects additional soft edges on the original images (Fig. 8 w4 and w5) which may be considered correct. However, TA-Loss provides the blurriest results, which is due to the regularization effect of applying the LTA detection during training.

The quantitative evaluation results are summarized in Table 1. The edge thresholds were set to 0.9 for **Linknet**, **RCF**, and **U-Net** and to 0.4, 0.5 and 0.6 for **Baseline**, **TA-Loss** and **TA-Correct**, respectively. These values were selected to get the best result for each method respectively. The proposed **TA-Loss** outperforms **Baseline** for most of the region and topology metrics. The proposed **TA-Correct** achieved the best performance for most of the metrics. It is noted that the baseline outperforms our approaches in some of the edge-based metrics, which is expected since the loss function is edge-based. We also want to point out that **U-Net** performs comparably to **TA-Correct** across the edge and region metrics but is significantly worse in the topology metric. Where many of the networks fail to close edge gaps, U-net provides more edges. The extra edges result in regions

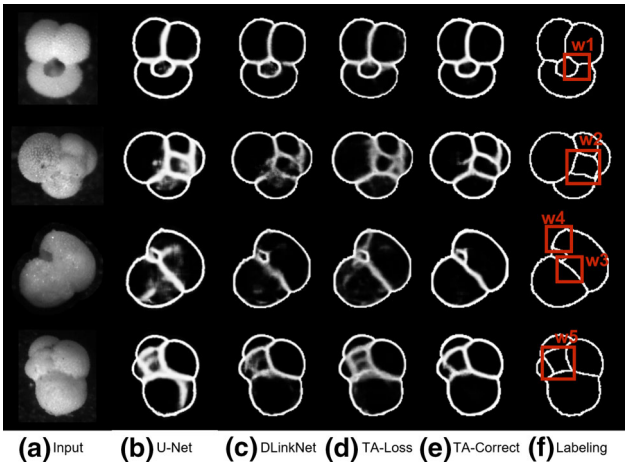


Fig. 8 Examples of resulting predictions. **a** Input image. **b U-Net** output. **c Baseline** output. **d TA-Loss** output. **e TA-Correct** output. **f** Groundtruth annotations. The red boxes highlight the regions the proposed approaches obtain clearer and more closed edges than the **Baseline** (color figure online)

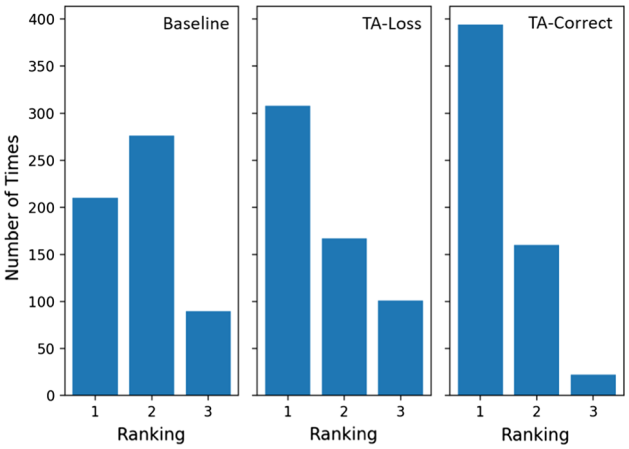


Fig. 9 Aggregate ranking results of expert and novices. Each bar counts how many times this method is ranked by the corresponding ranking value. The ranking for each participant showed the same trend as the aggregate

being closed, but additional edges within the region exist that do not create new regions as can be seen in Fig. 8. However, as discussed in the introduction, region and topology-based metric better capture the information that we care about for the morphological analysis of forams.

There is one method in Table 1 which has not been mentioned thus far. We added a **TA-Correct-CNN** method which replaces the CVAE in the **TA-Correct** method. Thus from Fig. 4, we have removed parts (b) and (c) and replaced them with a traditional CNN. By doing this, we show the effectiveness of the CVAE in reconciling topological differences. While the CNN may perform well pixel-wise, we have reviewed the effect of a generative model in reducing the "blurring" of logits versus a CNN in Sect. 3.3.2. As such, we hope to provide some motivation for using generative models for achieving such results.

To compare the quality of the networks with the top topological performance (i.e., **Baseline**, **TA-Loss** and **TA-Correct**), we had three researchers that perform studies with forams (one expert and two student novices) rank the three segmentation results for each of the 192 test images. During the ranking, each researcher was shown the image of the foram and the groundtruth segmentation, and the results of the three prediction methods in a random order for each image. They were asked to rank the results with values 1 to 3, where a value of 1 indicates the method provided the best segmentation. Ties were allowed, so there may be mul-

Table 1 Evaluation of the methods used for refinement

Metric	Linknet	RCF	U-Net	Baseline	TA-Loss	TA-Correct	TA-Correct-CNN
Edge							
Correct.	0.9260	0.9310	0.9419	*0.9437*	0.9395	**0.9533**	0.9375
Complete.	0.9324	0.9367	0.9472	**0.9494**	0.9482	*0.9493*	0.9493
Quality	0.8693	0.8767	0.8965	*0.8995*	0.8951	**0.9080**	0.8928
Region							
IoU	0.7403	0.7191	**0.7499**	0.7372	0.7372	*0.7481*	0.7480
W-IoU	0.8096	0.7692	**0.8283**	0.8021	0.8094	0.8130	*0.8242*
Recall (0.5)	0.7512	0.7114	0.8007	0.7682	0.7832	*0.8107*	**0.8161**
Recall (0.7)	0.6412	0.5749	*0.6904*	0.6460	0.6692	**0.7027**	0.7057
Topology							
Correct	79.67	72.79	83.10	86.24	*87.43*	**88.29**	79.39
Infeasible	1.26	3.53	0.39	0.18	0.17	*0.09*	**0.01**
LongShort	19.07	23.68	16.51	13.59	*12.40*	**11.62**	20.52

Correct., *Complete.* and *LongShort* denote the metrics *Correctness*, *Completeness* and *2Long2Short* mentioned in Sect. 4.4. The bold numbers indicate the method with the best performance for each metric with italics indicating second best scores

tiple rank 1 or 2 segmentation approaches for a single image. Figure 9 shows the aggregate counts of ranking values for all the researcher. The trends for each individual were similar. The **TA-Correct** has the most results ranked as 1 and the least results ranked as 3. The **TA-Loss** has more results ranked as 1 than the **Baseline**, but has about the same number of results ranked as 3, which indicates that the performance improvement of **TA-Loss** is less consistent than **TA-Correct**. More concretely, **TA-Loss** was ranked as high or higher than **Baseline** 67% of the time. **TA-Correct** was ranked as high or higher than **TA-Loss** 72% of the time. **TA-Correct** was ranked as high or higher than **Baseline** 90% of the time.

5 Conclusion and future work

We developed a detector of localized errors in edge maps based on localized homology conditions, and employ it to develop strategies that enforce preservation of topological structure in edge prediction models. The proposed strategies (TA-Loss and TA-Correct) showed significant improvement on the morphological segmentation of forams from the **NCSU-CUB Foram** dataset. We demonstrated improvement in region-based and topology-based metrics as well as determining that an expert and two novices consistently rank the segmentation by our approaches better than the baseline. Our experiments demonstrated that our refinement methodology for TA-Correct, in which a error detector and a region correction networks are added, does identify issues in the edge map and enhances the results.

In the future, we plan to enhance the LTA detection by considering a range of K_d values in order to provide a more gradual weighting of the loss. This will require developing more computationally-efficient algorithms that could make use of hierarchical structure as well as topological persistence. Also, we will explore techniques for end-to-end training of the TA-Correct approach.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://mwww.tensorflow.org. Accessed 27 Oct 2020.

Ahmed M., Fasy B. T., & Wenk C. (2014). Local persistent homology based distance between maps. In *Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems, SIGSPATIAL '14* (pp. 43–52). New York, NY: ACM.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916.

AROS Lab, NCSU. (2020). *NCSU-CUB ForaBot Project*. https://research.ece.ncsu.edu/aros/foram-identification/.

Bansal, A., Chen, X., Russell, B. C., Gupta, A., & Ramanan, D. (2017). *Pixelnet: Representation of the pixels, by the pixels, and for the pixels*. CoRR, arXiv: 1702.06506.

Beksi, W. J., & Papanikolopoulos, N. (2016). 3D region segmentation using topological persistence. In *2016 IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 1079–1084).

BenTaieb, A., & Hamarneh, G. (2016). Aware fully convolutional networks for histology gland segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 460–468). Berlin: Springer.

Berggren, W. A. (1972). A cenozoic time-scale-some implications for regional geology and paleobiogeography. *Lethaia*, *5*(2), 195–215.

Berggren, W. A. (1992). Ecology and palaeoecology of benthic foraminifera. *The Journal of Protozoology*, *39*(4), 537.

Bertasius, G., Shi, J., & Torresani, L. (2015). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *2015 IEEE Conference on computer vision and pattern recognition (CVPR)* (pp. 4380–4389). IEEE.

Boltovskoy, E., Scott, D. B., & Medioli, F. (1991). Morphological variations of benthic foraminiferal tests in response to changes in ecological parameters: A review. *Journal of Paleontology*, *65*(02), 175–185.

Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, *16*(1), 77–102.

Chaurasia, A., & Culurciello, E. (2017). Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual communications and image processing (VCIP)* (pp. 1–4). IEEE.

Chen, L., Barron, J. T., Papandreou, G., Murphy, K., & Yuille, A. L. (2015). *Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform*. CoRR, arXiv: 1511.03328.

Choi, C., & Christensen, H. I. (2012). 3D textureless object detection and tracking: An edge-based approach. In *2012 IEEE/RSJ International conference on intelligent robots and systems* (pp. 3877–3884).

Clough, J. R., Öksüz, I., Byrne, N., Schnabel, J. A., & King, A. P. (2019). *Explicit topological priors for deep-learning based image segmentation using persistent homology*. CoRR, arXiv: 1901.10244.

Corliss, B. H. (1991). Morphology and microhabitat preferences of benthic foraminifera from the northwest Atlantic Ocean. *Marine Micropaleontology*, *17*(3–4), 195–236.

Deng, J., Dong, W., Socher, R., Li, L., Kai, L., & Li F.-F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on computer vision and pattern recognition* (pp. 248–255).

Dollár, P., Tu, Z., & Belongie, S. (2006). Supervised learning of edges and object boundaries. In *2006 IEEE Computer Society conference on computer vision and pattern recognition (CVPR'06)* (Vol. 2, pp. 1964–1971).

Dollár, P., & Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(8), 1558–1570.

Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2000). Topological persistence and simplification. In *41st Annual symposium on foundations of computer science, 2000. Proceedings* (pp. 454–463). IEEE.

Ferrari, V., Fevrier, L., Jurie, F., & Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(1), 36–51.

Ge, Q., Zhong, B., Kanakiya, B., Mitra, R., Marchitto, T., & Lobaton, E. (2017). Coarse-to-fine foraminifera image segmentation through 3D and deep features. In *2017 IEEE Symposium series on computational intelligence (SSCI)* (pp. 1–8). IEEE.

Girshick, R. B. (2015). *Fast R-CNN*. CoRR, arXiv: 1504.08083.

Hu, X., Li, F., Samaras, D., & Chen, C. (2019). *Topology-preserving deep image segmentation*. CoRR, arXiv: 1906.05404.

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., & Keutzer, K. (2014). Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.

Kennett, J., & Srinivasan, M. (1983). *Neogene planktonic foraminifera: A phylogenetic atlas*. Stroudsburg: Hutchinson Ross.

Kingma , D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma , D. P., & Welling, M. (2014). Auto-encoding variational bayes. arXiv:1312.6114.

Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., & Rother, C. (2016). *Instancecut: From edges to instances with multicut*. CoRR, arXiv:1611.08272.

Letscher, D., & Fritts, J. (2007). Image segmentation using topological persistence. In W. G. Kropatsch, M. Kampel, & A. Hanbury (Eds.), *Computer analysis of images and patterns* (pp. 587–595). Berlin: Springer.

Lim, J. J., Zitnick, C. L., & Dollar, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Lin, T., Goyal, P., Girshick, R. B., He, K., & Dollár, P. (2017). *Focal loss for dense object detection*. CoRR, arXiv:1708.02002.

Liu, Y., Cheng, M., Hu, X., Bian, J., Zhang, L., Bai, X., et al. (2019). Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(8), 1939–1946.

Liu, Y., & Lew, M. S. (2016). Learning relaxed deep supervision for better edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 231–240).

Lobaton, E., Vasudevan, R., Bajcsy, R., & Alterovitz, R. (2010). Local occlusion detection under deformations using topological invariants. In *European conference on computer vision (ECCV)*.

Lobaton, E., Vasudevan, R., Bajcsy, R., & Alterovitz, R. (2011). Robust topological features for deformation invariant image matching. In *International conference on computer vision (ICCV)*.

Mnih, V., Heess, N., Graves, A., & Kavukcuoglu, K. (2014). Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberge (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 2204–2212). http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.pdf.

Mosinska, A., Márquez-Neila, P., Kozinski, M., & Fua, P. (2018). Beyond the pixel-wise loss for topology-aware delineation. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Qu, G., Zhang, W., Wang, Z., Dai, X., Shi, J., He, J., et al. (2018). Stripnet: Towards topology consistent strip structure segmentation. In *2018 ACM Multimedia conference on multimedia conference* (pp. 283–291). ACM.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement.

Rohling, E. J., & Cooke, S. (1999). Stable oxygen and carbon isotopes in foraminiferal carbonate shells. In *Modern foraminifera* (pp. 239–258).

Rojas-Moraleda, R., Xiong, W., Halama, N., Breitkopf-Heinlein, K., Dooley, S., Salinas, L., et al. (2017). Robust detection and segmentation of cell nuclei in biomedical images based on a computational topology framework. *Medical Image Analysis*, *38*, 90–103.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Berlin: Springer.

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761–769).

Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28, pp. 3483–3491). Boston: Curran Associates Inc.

Tipsword, H. L. (1962). Tertiary foraminifera in Gulf Coast petroleum exploration and development. In *Geology of the Gulf Coast and Central Texas, and guidebook of excursions* (pp. 16–57). http://archives.datapages.com/data/hgssp/data/013/013001/i_hgs013i.htm.

Walker, J., Doersch, C., Gupta, A., & Hebert, M. (2016). *An uncertain future: Forecasting from static images using variational autoencoders*. CoRR, arXiv:1606.07873.

Wegner, J. D., Montoya-Zegarra, J. A., & Schindler, K. (2013). A higher-order CRF model for road network extraction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1698–1705).

Wiedemann, C., Heipke, C., Mayer, H., & Jamet, O. (1998). Empirical evaluation of automatically extracted road axes. In *Empirical evaluation techniques in computer vision* (pp. 172–187).

Xie, S., & Tu, Z. (2017). Holistically-nested edge detection. *International Journal of Computer Vision*, *125*(1–3), 3–18.

Xu, D., Ouyang, W., Alameda-Pineda, X., Ricci, E., Wang, X., & Sebe, N. (2017). Learning deep structured multi-scale features using attention-gated CRFs for contour prediction. In I. Guyon , U. V. Luxburg , S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 3961–3970). http://papers.nips.cc/paper/6985-learning-deep-structured-multi-scale-features-using-attention-gated-crfs-for-contour-prediction.pdf.

Zhong, B., Ge, Q., Kanakiya, B., Marchitto, R. M. T., & Lobaton, E. (2017). A comparative study of image classification algorithms for foraminifera identification. In *2017 IEEE Symposium series on computational intelligence (SSCI)* (pp. 1–8). IEEE.

Zhou, L., Zhang, C., & Wu, M. (2018). D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*.

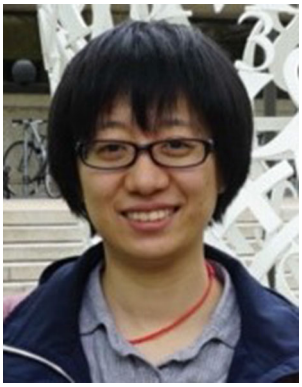
Zhu, G., Porikli, F., & Li, H. (2015). *Tracking randomly moving objects on edge box proposals*. CoRR, arXiv:1507.08085.

Zhu, Z., Xia, Y., Shen, W., Fishman, E., & Yuille, A. (2018). A 3D coarse-to-fine framework for volumetric medical image segmentation. In *2018 International conference on 3D vision (3DV)* (pp. 682–690). IEEE.

Zitnick, L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV*.

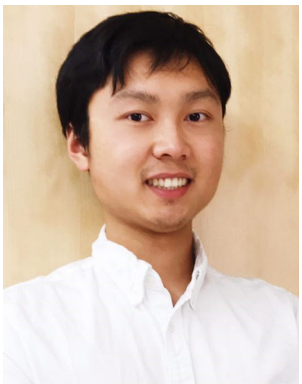
Zomorodian, A., & Carlsson, G. (2005). Computing persistent homology. *Discrete & Computational Geometry*, *33*(2), 249–274.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Qian Ge received B.S. and M.S. degree in Electrical Engineering from University of Electronic Science and Technology of China. She completed her Ph.D. in Electrical Engineering in 2019 from North Carolina State University. During her Ph.D. study, she worked with Dr. Edgar Lobaton at Active Robotic Sensing (ARoS) Laboratory on computer vision and machine learning. Her research interests are robust image segmentation using deep learning and topological data analysis tools.

Turner Richmond is a Ph.D. student advised by Dr. Lobaton in the Active Robotic Sensing Lab at North Carolina State University. He received his B.S. degree in Computer Science from University of North Carolina. Prior to joining NCSU, he worked as a software developer on projects supporting MBS trading. His research focuses include pattern recognition, computer vision, and computational geometric topology. His work at NCSU consists of contributing work on analysis of wearable sensors as well as species classification of microscopic fossils using topological properties of the species.



Boxuan Zhong received the B.E. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree in electrical and computer engineering with North Carolina State University, Raleigh, NC, USA. His current research interests include computer vision, machine learning, and robotics.

Thomas Marchitto is a Professor in the Department of Geological Sciences and a Fellow of the Institute of Arctic and Alpine Research, at the University of Colorado Boulder. He is a paleoceanographer, studying large-scale changes in ocean circulation and biogeochemistry that occur over timescales ranging from a few years to millions of years. Such changes are reflected in the physical and chemical properties of seawater, including temperature, salinity, carbonate chemistry, radiocarbon age, and the concentrations of various nutrients. Marchitto uses the chemistry of marine calcifiers, mainly foraminifera, as recorders of these properties.

Edgar J. Lobaton has been an Associate Professor in the Department of Electrical and Computer Engineering at North Carolina State University (NCSU) since 2011. Dr. Lobaton earned his B.S. in Mathematics and Electrical engineering from Seattle University in 2004. He completed his Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 2009. Dr. Lobaton was engaged in research at Alcatel-Lucent Bell Labs in 2005 and 2009. He was awarded the NSF CAREER Award in 2016. He was also awarded the 2009 Computer Innovation Fellows post-doctoral fellowship and conducted research in the Department of Computer Science at the University of North Carolina (UNC) at Chapel Hill from 2009 until 2011. His research focuses on the development of machine learning, estimation theory, and statistical and topological data analysis techniques applied to cyber-physical system in areas such wearable health monitoring, robotics and computer vision.