From Sparse to Precise: A Practical Editing Approach for Intracardiac Echocardiography Segmentation

Ahmed H. Shahin[1,2](✉), Yan Zhuang[1,3], and Noha El-Zehiry[1,4]

[1] Siemens Healthineers, New Jersey, USA
ahmedhshahen@gmail.com
[2] University College London, London, UK
[3] National Institutes of Health Clinical Center, Maryland, USA
[4] Wipro, New Jersey, USA

Abstract. Accurate and safe catheter ablation procedures for atrial fibrillation require precise segmentation of cardiac structures in Intracardiac Echocardiography (ICE) imaging. Prior studies have suggested methods that employ 3D geometry information from the ICE transducer to create a sparse ICE volume by placing 2D frames in a 3D grid, enabling the training of 3D segmentation models. However, the resulting 3D masks from these models can be inaccurate and may lead to serious clinical complications due to the sparse sampling in ICE data, frames misalignment, and cardiac motion. To address this issue, we propose an interactive editing framework that allows users to edit segmentation output by drawing scribbles on a 2D frame. The user interaction is mapped to the 3D grid and utilized to execute an editing step that modifies the segmentation in the vicinity of the interaction while preserving the previous segmentation away from the interaction. Furthermore, our framework accommodates multiple edits to the segmentation output in a sequential manner without compromising previous edits. This paper presents a novel loss function and a novel evaluation metric specifically designed for editing. Cross-validation and testing results indicate that, in terms of segmentation quality and following user input, our proposed loss function outperforms standard losses and training strategies. We demonstrate quantitatively and qualitatively that subsequent edits do not compromise previous edits when using our method, as opposed to standard segmentation losses. Our approach improves segmentation accuracy while avoiding undesired changes away from user interactions and without compromising the quality of previously edited regions, leading to better patient outcomes.

Keywords: Interactive editing · Ultrasound · Echocardiography

Work done while the authors were employed by Siemens Healthineers.
Disclaimer: The concepts and information presented in this paper are based on research results that are not commercially available. Future commercial availability cannot be guaranteed.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43901-8_73.

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14223, pp. 766–775, 2023.
https://doi.org/10.1007/978-3-031-43901-8_73

1 Introduction

Atrial Fibrillation (AFib) is a prevalent cardiac arrhythmia affecting over 45 million individuals worldwide as of 2016 [7]. Catheter ablation, which involves the elimination of affected cardiac tissue, is a widely used treatment for AFib. To ensure procedural safety and minimize harm to healthy tissue, Intracardiac Echocardiography (ICE) imaging is utilized to guide the intervention.

Intracardiac Echocardiography imaging utilizes an ultrasound probe attached to a catheter and inserted into the heart to obtain real-time images of its internal structures. In ablation procedures for Left Atrium (LA) AFib treatment, the ICE ultrasound catheter is inserted in the right atrium to image the left atrial structures. The catheter is rotated clockwise to capture image frames that show the LA body, the LA appendage and the pulmonary veins [12]. Unlike other imaging modalities, such as transesophageal echocardiography, ICE imaging does not require general anesthesia [3]. Therefore, it is a safer and more convenient option for cardiac interventions using ultrasound imaging.

The precise segmentation of cardiac structures, particularly the LA, is crucial for the success and safety of catheter ablation. However, segmentation of the LA is challenging due to the constrained spatial resolution of 2D ICE images and the manual manipulation of the ICE transducer. Additionally, the sparse sampling of ICE frames makes it difficult to train automatic segmentation models. Consequently, there is a persistent need to develop interactive editing tools to help experts modify the automatic segmentation to reach clinically satisfactory accuracy.

During a typical ICE imaging scan, a series of sparse 2D ICE frames is captured and a Clinical Application Specialist (CAS) annotates the boundaries of the desired cardiac structure in each frame[1] (Fig. 1a). To construct dense 3D masks for training segmentation models, Liao et al. utilized the 3D geometry information from the ICE transducer, to project the frames and their annotations onto a 3D grid [8]. They deformed a 3D template of the LA computed from 414 CT scans to align as closely as possible with the CAS contours, producing a 3D mesh to train a segmentation model [8]. However, the resulting mesh may not perfectly align with the original CAS contours due to factors such as frames misalignment and cardiac motion (Fig. 1b). Consequently, models trained with such 3D mesh as ground truth do not produce accurate enough segmentation results, which can lead to serious complications (Fig. 1c).

A natural remedy is to allow clinicians to edit the segmentation output and create a model that incorporates and follows these edits. In the case of ICE data, the user interacts with the segmentation output by drawing a scribble on one of the 2D frames (Fig. 1d). Ideally, the user interaction should influence the segmentation in the neighboring frames while preserving the original segmentation in the rest of the volume. Moreover, the user may make multiple edits to

[1] Annotations typically take the form of contours instead of masks, as the structures being segmented appear with open boundaries in the frames.

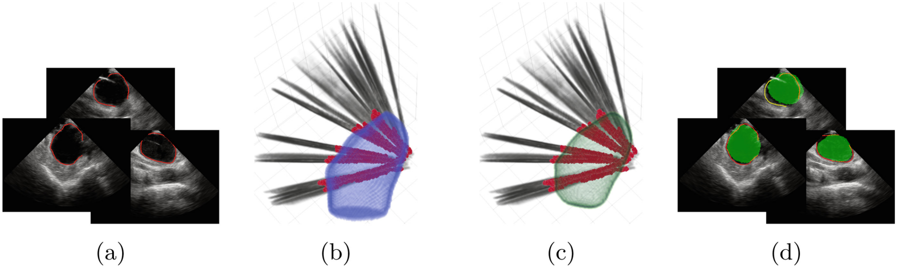

Fig. 1. Volumetric segmentation of ICE data. (a) 2D ICE frames with CAS contours outlining LA boundaries. (b) 2D frames (black shades) and CAS contours projected onto a 3D grid. The blue mesh represents the 3D segmentation mask obtained by deforming a CT template to fit the contours as closely as possible [8]. Note the sparsity of frames. (c) Predicted 3D segmentation mask generated by a model trained with masks from (b). (d) Predicted mask projected onto 2D (green) and compared with the original CAS contours. Note the misalignment between the mask and CAS contours in some frames. Yellow indicates an example of a user-corrective edit. (Color figure online)

the segmentation output, which must be incorporated in a sequential manner without compromising the previous edits.

In this paper, we present a novel interactive editing framework for the ICE data. This is the first study to address the specific challenges of interactive editing with ICE data. Most of the editing literature treats editing as an interactive segmentation problem and does not provide a clear distinction between interactive segmentation and interactive editing. We provide a novel method that is specifically designed for editing. The novelty of our approach is two-fold: 1) We introduce an editing-specific novel loss function that guides the model to incorporate user edits *while preserving the original segmentation in unedited areas.* 2) We present a novel evaluation metric that best reflects the editing formulation. Comprehensive evaluations of the proposed method on ICE data demonstrate that the presented loss function achieves superior performance compared to traditional interactive segmentation losses and training strategies, as evidenced by the experimental data.

2 Interactive Editing of ICE Data

2.1 Problem Definition

The user is presented first with an ICE volume, $x \in R^{H \times W \times D}$, and its initial imperfect segmentation, $y_{\text{init}} \in R^{H \times W \times D}$, where H, W and D are the dimensions of the volume. To correct inaccuracies in the segmentation, the user draws a scribble on one of the 2D ICE frames. Our goal is to use this 2D interaction to provide a 3D correction to y_{init} in the vicinity of the user interaction. We project the user interaction from 2D to 3D and encode it as a 3D Gaussian heatmap,

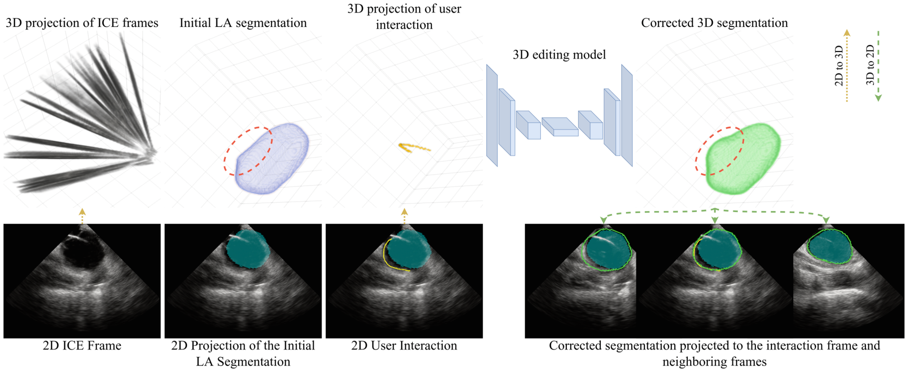
3D projection of ICE frames
Initial LA segmentation
3D projection of user interaction
3D editing model
Corrected 3D segmentation
2D to 3D
3D to 2D
2D ICE Frame
2D Projection of the Initial LA Segmentation
2D User Interaction
Corrected segmentation projected to the interaction frame and neighboring frames

Fig. 2. The proposed interactive editing framework involves user interaction with the segmentation output by drawing a scribble on one of the 2D frames. The editing model is trained to incorporate user interaction while preserving the initial segmentation in unedited areas. Cyan shade: initial segmentation. Green contour: corrected segmentation. Yellow contour: user interaction. (Color figure online)

$u \in R^{H \times W \times D}$, centered on the scribble with a standard deviation of σ_{enc} [9]. The user iteratively interacts with the output until they are satisfied with the quality of the segmentation.

We train an editing model f to predict the corrected segmentation output $\hat{y}^t \in R^{H \times W \times D}$ given x, y^t_{init}, and u^t, where t is the iteration number. The goal is for $\hat{y}^t$ to accurately reflect the user's correction near their interaction while preserving the initial segmentation elsewhere. Since $y^{t+1}_{\text{init}} \equiv \hat{y}^t$, subsequent user inputs $u^{\{t+1,\ldots,T\}}$ should not corrupt previous corrections $u^{\{0,\ldots,t\}}$ (Fig. 2).

2.2 Loss Function

Most interactive segmentation methods aim to incorporate user guidance to enhance the overall segmentation [2,5,9]. However, in our scenario, this approach may undesirably modify previously edited areas and may not align with clinical expectations since the user has corrected these areas, and the changes are unexpected. To address the former issue, Bredell et al. proposed an iterative training strategy in which user edits are synthesized and accumulated over a fixed number of steps with every training iteration [1]. However, this approach comes with a significant increase in training time and does not explicitly instruct the model to preserve regions away from the user input.

We propose an **editing-specific** loss function $\mathcal{L}$ that encourages the model to preserve the initial segmentation while incorporating user input. The proposed loss function incentivizes the model to match the prediction $\hat{y}$ with the ground truth y in the vicinity of the user interaction. In regions further away from the user interaction, the loss function encourages the model to match the initial segmentation y_{init}, instead. Here, y represents the 3D mesh, which is created by deforming a CT template to align with the CAS contours y_{cas} [8]. Meanwhile, y_{init} denotes the output of a segmentation model that has been trained on y.

We define the vicinity of the user interaction as a 3D Gaussian heatmap, $A \in R^{H\times W\times D}$, centered on the scribble with a standard deviation of σ_{edit}. Correspondingly, the regions far from the interaction are defined as $\bar{A} = 1 - A$. The loss function is defined as the sum of the weighted cross entropy losses $\mathcal{L}_{\text{edit}}$ and $\mathcal{L}_{\text{preserve}}$ w.r.t y and y_{init}, respectively, as follows

$$\mathcal{L} = \mathcal{L}_{\text{edit}} + \mathcal{L}_{\text{preserve}} \tag{1}$$

where

$$\mathcal{L}_{\text{edit}} = -\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{D} A_{i,j,k}\left[y_{i,j,k}\log\hat{y}_{i,j,k} + (1-y_{i,j,k})\log(1-\hat{y}_{i,j,k})\right] \tag{2}$$

$$\mathcal{L}_{\text{preserve}} = -\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{D} \bar{A}_{i,j,k}\left[y_{init_{i,j,k}}\log\hat{y}_{i,j,k} + (1-y_{init_{i,j,k}})\log(1-\hat{y}_{i,j,k})\right] \tag{3}$$

The Gaussian heatmaps facilitate a gradual transition between the edited and unedited areas, resulting in a smooth boundary between the two regions.

2.3 Evaluation Metric

The evaluation of segmentation quality typically involves metrics such as the Dice coefficient and the Jaccard index, which are defined for binary masks, or distance-based metrics, which are defined for contours [13]. In our scenario, where the ground truth is CAS contours, we use distance-based metrics[2]. However, standard utilization of these metrics computes the distance between the predicted and ground truth contours, which misleadingly incentivizes alignment with the ground truth contours in **all regions**. This approach incentivizes changes in the unedited regions, which is undesirable from a user perspective, as users want to see changes only in the vicinity of their edit. Additionally, this approach incentivizes the corruption of previous edits.

We propose a novel editing-specific evaluation metric that assesses how well the prediction $\hat{y}$ matches the CAS contours y_{cas} in the vicinity of the user interaction, and the initial segmentation y_{init} in the regions far from the interaction.

$$\mathcal{D} = \mathcal{D}_{\text{edit}} + \mathcal{D}_{\text{preserve}} \tag{4}$$

where, $\forall (i,j,k) \in \{1,\ldots,H\} \times \{1,\ldots,W\} \times \{1,\ldots,D\}$, $\mathcal{D}_{\text{edit}}$ is the distance from y_{cas} to $\hat{y}$ in the vicinity of the user edit, as follows

$$\mathcal{D}_{\text{edit}} = \mathbb{1}_{(y_{\text{cas}_{i,j,k}}=1)} \cdot A_{i,j,k} \cdot d(y_{\text{cas}_{i,j,k}}, \hat{y}) \tag{5}$$

[2] Contours are inferred from the predicted mask $\hat{y}$.

where d is the minimum Manhattan distance from $y_{\text{cas}_{i,j,k}}$ to any point on $\hat{y}$. For $\mathcal{D}_{\text{preserve}}$, we compute the average symmetric distance between y_{init} and $\hat{y}$, since the two contours are of comparable length. The average symmetric distance is defined as the average of the minimum Manhattan distance from each point on y_{init} contour to $\hat{y}$ contour and vice versa, as follows

$$\mathcal{D}_{\text{preserve}} = \frac{\bar{A}}{2} \cdot \left[\mathbb{1}_{(y_{\text{init}_{i,j,k}}=1)} \cdot d(y_{\text{init}_{i,j,k}}, \hat{y}) + \mathbb{1}_{(\hat{y}_{i,j,k}=1)} \cdot d(\hat{y}_{i,j,k}, y_{\text{init}}) \right] \tag{6}$$

The resulting $\mathcal{D}$ represents a distance map $\in R^{H \times W \times D}$ with defined values only on the contours $y_{\text{cas}}, y_{\text{init}}, \hat{y}$. Statistics such as the 95^{th} percentile and mean can be computed on the corresponding values of these contours on the distance map.

3 Experiments

3.1 Dataset

Our dataset comprises ICE scans for 712 patients, each with their LA CAS contours y_{cas} and the corresponding 3D meshes y generated by [8]. Scans have an average of 28 2D frames. Using the 3D geometry information, frames are projected to a 3D grid with a resolution of $128 \times 128 \times 128$ and voxel spacing of $1.1024 \times 1.1024 \times 1.1024$ mm. We performed five-fold cross-validation on 85% of the dataset (605 patients) and used the remaining 15% (107 patients) for testing.

3.2 Implementation Details

To obtain the initial imperfect segmentation y_{init}, a U-Net model [11] is trained on the 3D meshes y using a Cross-Entropy (CE) loss. The same U-Net architecture is used for the editing model. The encoding block consists of two 3D convolutional layers followed by a max pooling layer. Each convolutional layer is followed by batch normalization and ReLU non-linearity layers [4]. The number of filters in the segmentation model convolutional layers are 16, 32, 64, and 128 for each encoding block, and half of them for the editing model. The decoder follows a similar architecture.

The input of the editing model consists of three channels: the input ICE volume x, the initial segmentation y_{init}, and the user input u. During training, the user interaction is synthesized on the frame with maximum error between y_{init} and y.[3] The region of maximum error is selected and a scribble is drawn on the boundary of the ground truth in that region to simulate the user interaction. During testing, the real contours of the CAS are used and the contour with the maximum distance from the predicted segmentation is chosen as the user interaction. The values of σ_{enc} and σ_{edit} are set to 20, chosen by cross-validation. Adam optimizer is used with a learning rate of 0.005 and a batch size of 4 to train the editing model for 100 epochs [6].

[3] We do not utilize the CAS contours during training and only use them for testing because the CAS contours do not align with the segmentation meshes y.

Table 1. Results on Cross-Validation (CV) and test set. We use the editing evaluation metric $\mathcal{D}$ and report the 95$^{\text{th}}$ percentile of the overall editing error, the error near the user input, and far from the user input (mm). The near and far regions are defined by thresholding A at 0.5. For the CV results, we report the mean and standard deviation over the five folds. The statistical significance is computed for the difference with InterCNN. †: p-value < 0.01, ‡: p-value < 0.001.

Method	CV			Test		
	Overall ↓	Near ↓	Far ↓	Overall ↓	Near ↓	Far ↓
No Editing	3.962 ± 0.148	-	-	4.126	-	-
CE Loss	1.164 ± 0.094	0.577 ± 0.024	0.849 ± 0.105	1.389	0.6	1.073
Dice Loss	1.188 ± 0.173	0.57 ± 0.089	0.892 ± 0.155	1.039	**0.46**	0.818
InterCNN	0.945 ± 0.049	**0.517 ± 0.052**	0.561 ± 0.006	0.94	0.509	0.569
Editing Loss	**0.809 ± 0.05**‡	0.621 ± 0.042	**0.182 ± 0.01**‡	**0.844**†	0.662	**0.184**‡

3.3 Results

We use the editing evaluation metric $\mathcal{D}$ (Sect. 2.3) for the evaluation of the different methods. For better interpretability of the results, we report the overall error, the error near the user input, and the error far from the user input. We define near and far regions by thresholding the Gaussian heatmap A at 0.5.

We evaluate our loss (editing loss) against the following baselines: (1) **No Editing**: the initial segmentation y_{init} is used as the final segmentation $\hat{y}$, and the overall error in this case is the distance from the CAS contours to y_{init}. This should serve as an upper bound for error. (2) **CE Loss**: an editing model trained using the standard CE segmentation loss w.r.t y. (3) **Dice Loss** [10]: an editing model trained using Dice segmentation loss w.r.t y. (4) **InterCNN** [1]: for every training sample, simulated user edits based on the prediction are accumulated with any previous edits and re-input to the model for 10 iterations, trained using CE loss. We report the results after a single edit (the furthest CAS contour from $\hat{y}$) in Table 1. A single training epoch takes ≈ 3 min for all models except InterCNN, which takes ≈ 14 min, on a single NVIDIA Tesla V100 GPU. The inference time through our model is ≈ 20 milliseconds per volume.

Our results demonstrate that the proposed loss outperforms all baselines in terms of overall error. Although all the editing methods exhibit comparable performance in the near region, in the far region where the error is calculated relative to y_{init}, our proposed loss outperforms all the baselines by a significant margin. This can be attributed to the fact that the baselines are trained using loss functions which aim to match the ground truth globally, resulting in deviations from the initial segmentation in the far region. In contrast, our loss takes into account user input in its vicinity and maintains the initial segmentation elsewhere.

Sequential Editing. We also investigate the scenario in which the user iteratively performs edits on the segmentation multiple times. We utilized the same models that were used in the single edit experiment and simulated 10 editing iterations. At each iteration, we selected the furthest CAS contour from $\hat{y}$,

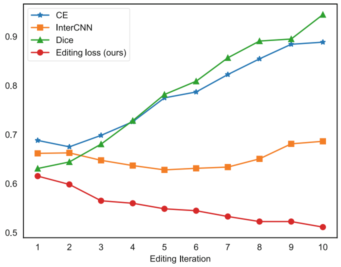

Fig. 3. 95th percentile of the distance from the CAS contours to the prediction.

ensuring that the same edit was not repeated twice. For the interCNN model, we aggregated the previous edits and input them into the model, whereas for all other models, we input a single edit per iteration. We assessed the impact of the number of edits on the overall error. In Fig. 3, we calculated the distance from all the CAS contours to the predicted segmentation and observed that the editing loss model improved with more edits. In contrast, the CE and Dice losses degraded with more edits due to compromising the previous corrections, while InterCNN had only marginal improvements.

Furthermore, in Fig. 4, we present a qualitative example to understand the effect of follow-up edits on the first correction. Edits after the first one are on other frames and not shown in the figure. We observe that the CE and InterCNN methods did not preserve the first correction, while the editing loss model maintained it. This is a crucial practical advantage of our loss, which allows the user to make corrections without compromising the previous edits.

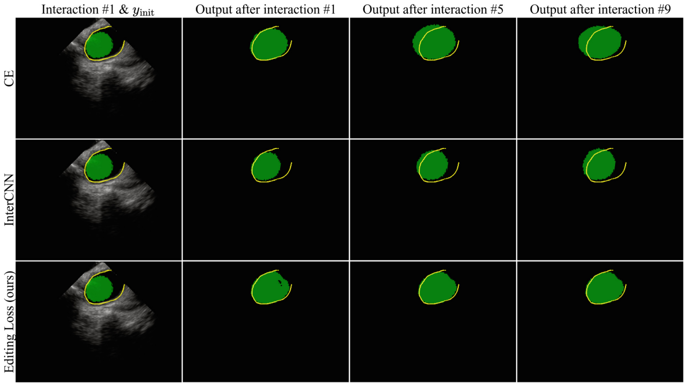
Interaction #1 & y_{init}
Output after interaction #1
Output after interaction #5
Output after interaction #9
CE
InterCNN
Editing Loss (ours)

Fig. 4. Impact of follow-up edits on the first correction. Yellow: first user edit. Green: output after each edit. Our editing loss maintains the integrity of the previous edits, while in the other methods the previous edits are compromised. (Color figure online)

4 Conclusion

We presented an interactive editing framework for challenging clinical applications. We devised an editing-specific loss function that penalizes the deviation from the ground truth near user interaction and penalizes deviation from the initial segmentation away from user interaction. Our novel editing algorithm is more robust as it does not compromise previously corrected regions. We demonstrate the performance of our method on the challenging task of volumetric segmentation of sparse ICE data. However, our formulation can be applied to other editing tasks and different imaging modalities.

References

1. Bredell, G., Tanner, C., Konukoglu, E.: Iterative interaction training for segmentation editing networks. In: Shi, Y., Suk, H.I., Liu, M. (eds.) Machine Learning in Medical Imaging, pp. 363–370 (2018)
2. Dorent, R., et al.: Inter extreme points geodesics for end-to-end weakly supervised image segmentation. In: de Bruijne, M., et al. (eds.) MICCAI 2021. LNCS, vol. 12902, pp. 615–624. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87196-3_57
3. Enriquez, A., et al.: Use of intracardiac echocardiography in interventional cardiology: working with the anatomy rather than fighting it. Circulation **137**(21), 2278–2294 (2018)
4. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
5. Khan, S., Shahin, A.H., Villafruela, J., Shen, J., Shao, L.: Extreme points derived confidence map as a cue for class-agnostic interactive segmentation using deep neural network. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11765, pp. 66–73. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32245-8_8
6. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Kornej, J., Börschel, C.S., Benjamin, E.J., Schnabel, R.B.: Epidemiology of atrial fibrillation in the 21st century. Circ. Res. **127**(1), 4–20 (2020)
8. Liao, H., Tang, Y., Funka-Lea, G., Luo, J., Zhou, S.K.: More knowledge is better: cross-modality volume completion and 3D+2D segmentation for intracardiac echocardiography contouring. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11071, pp. 535–543. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_60
9. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: from extreme points to object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 616–625 (2018)
10. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571 (2016)
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

12. Russo, A.D., et al.: Role of intracardiac echocardiography in atrial fibrillation ablation. J. Atr. Fibrillation **5**(6) (2013)
13. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med. Imaging **15**(1), 1–28 (2015)