This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3186146

Finding Nano-Ötzi: Cryo-Electron Tomography Visualization Guided by Learned Segmentation

Ngan Nguyen*, Ciril Bohak*, Dominik Engel, Peter Mindek, Ondřej Strnad,
Peter Wonka, Sai Li, Timo Ropinski, Ivan Viola

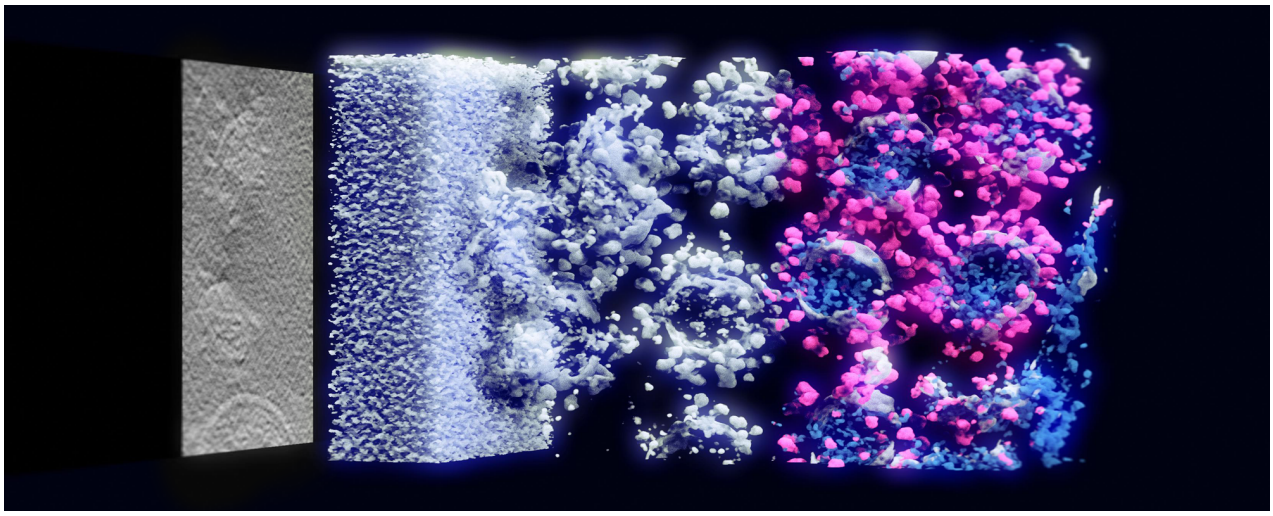

Fig. 1: The image shows a volume containing several intact SARS-CoV-2 virions acquired using cryo-electron tomography 3D imaging. From left to right: slice of the original data; direct volume rendering of the original data; foreground-background segmentation; color-coded four-class segmented data (background, spikes, membrane, lumen).

Abstract—Cryo-electron tomography (cryo-ET) is a new 3D imaging technique with unprecedented potential for resolving submicron structural details. Existing volume visualization methods, however, are not able to reveal details of interest due to low signal-to-noise ratio. In order to design more powerful transfer functions, we propose leveraging soft segmentation as an explicit component of visualization for noisy volumes. Our technical realization is based on semi-supervised learning, where we combine the advantages of two segmentation algorithms. First, the weak segmentation algorithm provides good results for propagating sparse user-provided labels to other voxels in the same volume and is used to generate dense pseudo-labels. Second, the powerful deep-learning-based segmentation algorithm learns from these pseudo-labels to generalize the segmentation to other unseen volumes, a task that the weak segmentation algorithm fails at completely. The proposed volume visualization uses deep-learning-based segmentation as a component for segmentation-aware transfer function design. Appropriate ramp parameters can be suggested automatically through frequency distribution analysis. Furthermore, our visualization uses gradient-free ambient occlusion shading to further suppress the visual presence of noise, and to give structural detail the desired prominence. The cryo-ET data studied in our technical experiments are based on the highest-quality tilted series of intact SARS-CoV-2 virions. Our technique shows the high impact in target sciences for visual data analysis of very noisy volumes that cannot be visualized with existing techniques.

Index Terms—Volume Rendering; Computer Graphics Techniques; Machine Learning Techniques; Scalar Field Data; Life Sciences

1 Introduction

Since 2014, due to the revolution in resolution [1], cryogenic electron microscopy (cryo-EM) has been the main technique for high-resolution macromolecule structure determination. Its extension cryo-electron tomography (cryo-ET)

- * *shared first authorship*
- *Ngan N., Ciril B., Ondřej S., Peter W., and Ivan V. - King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. E-mail: {ngan.nguyen | ciril.bohak | ondrej.strnad | peter.wonka | ivan.viola }@kaust.edu.sa.*
- *Dominik E., and Timo R. - Ulm University, Ulm, Germany. E-mail: {dominik.engel | timo.ropinski }@uni-ulm.de.*
- *Peter M. - TU Wien and Nanographics GmbH. E-mail: mindek@cg.tuwien.ac.at.*
- *Sai L. - Tsinghua University School of Life Sciences, Bejing, China. E-mail: sai@tsinghua.edu.cn*

Manuscript received xx xxx, 202x; revised xx xxx, 202x.

enables the use of cryo-EM for the 3D reconstruction of specimens. As the technique has become more widely accessible, the amount of acquired data has far superseded the existing data analysis pipelines' capacities. While acquisitions are still mostly performed manually by microbiology experts, with the rapid increase in acquired data, new analysis tools need to be developed.

One crucial analysis step is determining which specimen is represented in the acquired data and how to proceed with its processing. This is typically carried out using 3D data visualization methodologies. However, imaging artifacts can make this quite challenging, and is analogous to discovering fossils at an excavation site. Due to their fossilization, specimens have almost the same composition as the surrounding environment and are difficult to distinguish. The interesting details in cryo-ET data are similarly *buried* in the surrounding noise. Since the noise is a direct result of the acquisition process, where the energized particles must be evenly spread so that they cause as little damage to the specimen as possible, cryo-ET data always suffers from a low signal-to-noise ratio (SNR).

To *excavate* the specimens from the surrounding noise, experts typically annotate the data manually and use those annotations for further steps in their research (*e.g.*, subtomogram averaging [2] to determine the detailed structure composition or visualization). As this process can only be completed by domain experts, data processing has become a bottleneck and researchers are looking into modern automatic segmentation methods based on deep learning (DL). DL-based semantic segmentation approaches have shown great potential in a wide range of scientific disciplines, such as medicine [3] and biology [4]. However, when applying standard discrete DL-based semantic segmentation to cryo-ET data, the obtained crisp segmentation masks do not accurately reflect the uncertainty stemming from the imaging modality. Rather than having each pixel classified into a distinct—but possibly wrong—class, the low resolution and SNR require a higher degree of flexibility in exploring the data by the domain experts. It would be desirable to use a transfer function (TF), as is possible when exploring the medical volume data, but this is not feasible due to the low SNR of cryo-ET data.

This technical contribution is centered around the key observation that visual mapping using a TF specification essentially performs two tasks: (1) TF performs a *soft segmentation* of objects and, simultaneously, (2) TF *assigns optical properties*. Solving both tasks at the same time is non-trivial, even for *easy* noise-free modalities like medical CT data. However, by decomposing the role of visual mapping into two separate tasks, a solution that allows a high degree of automation, even for the most challenging and noise-polluted modalities such as cryo-ET, can be found .

We translate the task of soft segmentation during the visual mapping stage into the domain of probabilistic segmentation, which results in the desired soft membership assignment. To achieve this, we adopt the concept of *semi-supervised learning* within the volume visualization pipeline. This methodology provides high-quality, soft segmentation even from sparse user input. With just a few supervising sparse strokes, we employ a weaker segmentation algorithm to generate dense labels, also called pseudo-labels. We provide these pseudo-labels to a stronger deep-learning classifier that is robust but more data-hungry, as it requires dense labels for training. Once trained, it provides a high-quality probabilistic (soft) segmentation for unseen datasets fully automatically.

The soft segmentation task is coupled with an optical properties assignment in direct volume rendering (DVR), which is much easier to automate when tackled separately. Together, these two steps form a visual mapping assignment which results, together with advanced volume illumination models, in rendered images of high visual quality. Finally, the 3D visualization can be fine-tuned by the user if the automated methods do not find the best exact visual parameters. We summarize our contributions as follows:

- We propose aiding users in designing TF by decomposing the visual mapping task, as suggested by Drebin *et al.* [5], empowered by a modern deep-learning approach.
- For the soft segmentation, we propose using the concept of semi-supervised learning based on sparse user input to employ a weak classifier for obtaining the pseudo-labels that serve as input for a strong classifier for obtaining the final labels.
- We propose combining the details of the raw data with the softness of the segmentation to estimate the opacity mapping using an iterative thresholding algorithm.
- We demonstrate and evaluate our concept for processing challenging cryo-ET data and gather expert feedback to assess the potential of the novel volume visualization pipeline.

Metaphorically, we relate our new visualization technique of the frozen specimen to the discovery of Ötzi, or Ice Man, the oldest natural mummy.

In the following section, we relate our proposed approach to state-of-the-art alternatives and prior work. Next, we give an overview of the proposed method and show how its individual components work together to yield a complete system and contribute to the final visualization. Sections 4 and 5 present the novel components of our approach in detail level and are followed by a demonstration of our results and their evaluation, where the results are discussed with two domain experts. In the final section, we conclude the paper and present possible future directions. Our code and pre-trained models are publicly available[1].

2 Related Work

Cryo-ET has evolved dramatically over recent years [6] and has become a go-to method for most high-resolution *in situ* structural biology challenges [7]. When combined with subtomogram averaging [8] to obtain greater detail of the studied structure, this is a perfect method for examining and analyzing the molecular architecture of new viruses, such as SARS-CoV-2 [9].

Due to the nature of the acquisition process, the SNR is meager. The reasons for this include an inability to create the perfect vacuum inside the acquisition tube—resulting in

1. https://github.com/nanovis/nano-oetzi

floating dust particles; imperfect acquisition sensors, which do not always produce a perfect image; inconsistencies in the preparation of the specimens; and limitations on the energy used during the acquisition process. These issues, amongst others, produce noise which obscures the specimen and lowers the SNR. Huang *et al.* [10] addressed this problem by optimizing wavelet-based filters. Shigematsu and Sigworth [11] analyzed different noise models. Both studies concluded that a Gaussian noise model—with preparation pipeline-dependent parameters—describes the noise in cryo-ET data best. While we could use a Gaussian filter to denoise the input data, this would also remove information from the high-frequency domain, which we want to avoid in our pipeline. The denoising process itself cannot be simply used to segment the structures of interest; we would still need to label the denoised data and use an additional segmentation method.

It is also proven that many existing denoising approaches based on deep learning [10], [11] are unsuited for cryo-ET data as they require *clean* targets without noise for training, which is currently impossible to acquire in this domain. For this reason, most existing denoising approaches are based on the Noise2Noise approach [12], which avoids this noise-free requirement by training on pairs of registered noisy data. The Topaz Denoise method [13] is an extension of the Noise2Noise method, which was adapted for cryo-EM data. While Noise2Noise was not developed with cryo-EM data in mind, such pairs of registered noisy cryo-EM images can be acquired using dose-fractioning, which splits the acquired electron dose in half, resulting in two independent images. Buchholz *et al.* [14] used this approach to acquire such data and proposed Cryo-Care, which builds on Noise2Noise and provides denoising of both cryo-EM images and tomograms. Noise2Void [15] also reports promising results for cryo-EM images. Su *et al.* [16] presented a generative adversarial network (GAN) trained using synthetic data as *clean* examples and synthetically degraded images as input to the denoiser, resulting in a model that can cope with different varieties of learned noise. In our case, we could not use Noise2Noise models, since we have neither registered noisy and denoised images nor noisy images obtained using dose-fractioning. While Noise2Void models could be used in the preprocessing step, annotation would still be required for the segmentation task. Since we did not want to lose any information, we omitted such a preprocessing step.

Deep learning approaches have also shown very promising results in segmentation tasks for both images [17], [18], [19], [20] and volumetric data [21], [22]. Most segmentation models are fully convolutional neural networks consisting of an encoder and a decoder, with skip connections between them. This encoder-decoder architecture is the basis for most segmentation networks, including the popular U-Net [23]. The U-Net, in particular, is the base of many segmentation architectures that followed. While most architectural innovation was pioneered in 2D [17], many of the successful approaches can be extended to 3D. Following this recipe, the 3D U-Net [21] was created by extending its 2D counterpart [23]. Lee *et al.* [24] proposed using residual blocks in U-Nets, in addition to anisotropic convolution kernels, to account for worse reconstruction quality in the z-axis. Both the 3D U-Net [21] and the residually symmetric U-Net [24] are considered in our initial architecture selection. Another relevant line of work is that of Bui *et al.* [25], [26], [27] and Yu *et al.* [28], who proposed different versions of a skip-connected 3D DenseNet for segmentation. We include its best-performing variant [27] in our initial architecture selection. Siddique *et al.* [29] further provide an overview of different U-Net and DenseNet variants applied to different problems in the medical domain. While many of the variants discussed in their work, such as attention U-Nets [30] or UNet++ variants [31], [32], are an improvement over the simple base variants, we chose to use the aforementioned three simple representatives of the most prominent directions in this space for our approach. This is also because most of the presented approaches do not evaluate the TEM modality, thus their superiority is not clearly demonstrated for our data. Most closely related to our approach, Moebel et al. [33] present a deep learning based approach to macromolecule localization in Cryo-EM data. Specifically they propose using a 3D variant of a U-Net for the segmentation step in their approach, before localizing clusters in the segmented volumes. Lastly, Gros *et al.* [34] proposed SoftSeg and investigated techniques to deal with non-binary segmentation labels. This approach is orthogonal to the above works. The authors proposed different activation functions for the output layer and adapted loss functions to deal with the soft labels. Using soft labels can be beneficial because they can elegantly incorporate uncertainty and inter-expert variability into the labeling process. In this work, we also make use of soft labels. Having uncertainty in our trained network's predictions enables us to visualize the data with its uncertainty accordingly.

Having discussed the denoising and segmentation approaches, we highlight here some related works that use deep learning approaches to replace parts of the visualization pipeline itself. Cheng *et al.* [35] proposed using a learned feature space for TF design instead of the raw intensity or first and/or second-order derivatives. Users can design TFs within a widget by choosing features relevant to them from an ordered feature list extracted by the neural network. DNN-VolVis [36] goes a step further and uses a neural network for the shading. Specifically, an unshaded image is rendered from the desired viewpoint and an image-to-image translation network applies shading in the style of an additionally supplied style image. Taking that a step further, Berger *et al.* [37] proposed a GAN that fully synthesizes the desired renderings based on only a viewpoint and a TF, leaving the whole rendering process to the neural network.

Clear DVR is only possible with a good definition of how the volume data translates into renderable optical properties, as defined by a TF. The process of TF design has been extensively researched. The first rule-based approach to defining the TF was proposed by Bergman *et al.* [38] and used to color meteorological and flow simulation volumetric data. Kindlman and Durkin [39] presented a semi-automatic approach for TF generation for visualizing material boundaries, taking into account intensities and their first and second derivatives. Correa and Ma [40] later introduced a semi-automated method for generating TFs in which they progressively explore TF space to maximize the visibility of important structures. Lindholm *et al.* [41] present an approach for transfer function design with spatial localization based on user specified material dependencies.

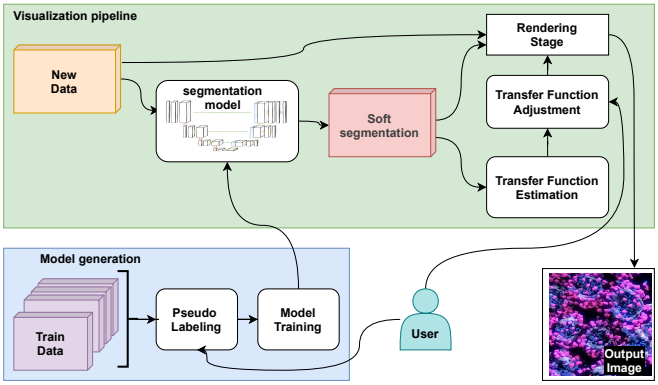

Fig. 2: System overview. First, sparse user-provided labels are propagated to obtain dense pseudo-labels. A deep-learning-based 3D segmentation model is trained on the pseudo-labeled data; the resulting soft segmentation is next used in the TF parameter estimation step, where ramping parameters are estimated. The TF can be further adjusted by the user before being used together with the raw data and soft segmentation in the rendering stage to produce the final visualization output.

Cai *et al.* [42] introduced automatic TF generation using visibility distributions and projective color mapping, which matches the distribution of visible values in the current view with a target one for equal pronouncement of all the features. Ljung *et al.* [43] presented a thorough overview of TFs for DVR and ongoing challenges. Luo and Dingliana [44] presented a TF optimization based on visibility and saliency. A recent study by Ma and Entezari [45] suggested the use of cell-based isosurface similarity, feature-based classification, and visibility analysis for a semi-automatic TF design.

Another closely related line of work is uncertainty visualization. When visualizing predicted or approximative data, it is desirable that the user is informed about the confidence of those predictions. Prassni *et al.* [46] visualized unconfident segmentations using sets of isolines, which naturally coincide in more certain regions, where there is a distinct line between the foreground and background. In uncertain regions, these lines spread out as the transition between the foreground and background is more gradual. In 3D visualization, they displayed uncertainty through a set of semi-transparent iso-surfaces. Lundström *et al.* [47] proposed an uncertainty-aware TF design. Their approach allows the definition of two separate 1D TFs, one for a fully certain prediction and another for a fully uncertain prediction. Depending on the actual degree of certainty of a sample classification, the two TFs are interpolated. In another work [48], the use of an animation to highlight uncertain regions in the classification is proposed. Diepenbrock *et al.* [49] directly lowered the saturation and value of standardized (HSV) color maps used to encode directions in fiber visualization to convey uncertainty.

3 Technical Overview

Our proposed approach enables semi-supervised DVR. We show that even 3D visualization of cryo-ET data becomes possible with this new approach, which is typically not achievable with current volume rendering systems due to the low SNR (see Figure 9). To achieve the desired high-quality visualization, we decompose the visual mapping stage into two sub-problems. One is automatic opacity mapping using the iterative thresholding algorithm and a mixture of a soft segmentation signal with a crisp raw-data signal. The second sub-problem is soft segmentation through a probabilistic approach using the semi-supervised learning methodology. Our probabilistic segmentation is composed of a DL inference that performs the automatic labeling *across* unseen datasets at runtime. This stage works well even on very challenging modalities; however, it is very training-data hungry. The probabilistic segmentation is therefore trained in the pre-training stage from dense labels.

We denote these input labels as *pseudo-labels* following the semi-supervised learning literature. While these labels characterize a particular volume well, the method that generates them *within* the volume performs badly when generalizing *across* volumes that contain similar structures. The advantage of the pseudo-labeling method that we employ is that it produces good results for assigning soft labels *within* a volume based on only sparse user input, but does not generalize across volumes. The other technique performs good segmentation *across* volumes but requires dense input. By integrating these two approaches, the sparse-input guided segmentation with training a DL classifier, we obtain a semi-supervised soft segmentation concept that proves beneficial in the context of visual mapping within the volume visualization pipeline. We present our pipeline for visual mapping of two classes, *i.e.*, foreground and background, or four classes where distinct structures of a viral specimen are distinguished by distinct colors. Finally, the volume visualization incorporates integrative illumination models that further amplify the visual presence of signal over noise.

As illustrated in Figure 2, the overall method consists of two parts. The first part is the model generation, which takes the sparse user-annotated labels as input and produces dense pseudo-labels. These pseudo-labels are then used by a DL-based segmentation algorithm to train a final 3D segmentation model. The second part is the visualization pipeline, which takes new data, probabilistically segments it into four classes, estimates the TF parameters, and renders the data.

4 Model Generation

The final result of the *model generation* step is a trained deep neural network for probabilistic semantic segmentation that can be used in our proposed visualization pipeline. To achieve this goal, we draw from concepts in semi-supervised learning [50], [51]. The semi-supervised learning setting applies to a situation where a smaller set of labeled data is available together with a (typically) larger set of unlabeled data. In our context, we have volumetric data. We are given a smaller set of manually labeled voxels $x_1, x_2, \ldots, x_l$ with labels $y_1, y_2, \ldots, y_l$ and a larger set of unlabeled voxels $x_{l+1}, x_{l+2}, \ldots, x_{l+u}$, where l is the number of labeled voxels and u is the number of unlabeled voxels. Specifically, in our setting, the labeled voxels are sparsely distributed within a given set of volumes. Since each volume is very large,

it would be too time-consuming to manually create voxel-precise labels even for a single volume. Our proposed solution is to use two different segmentation algorithms, leveraging the advantages of each of them. First, we use a weak segmentation algorithm provided by a state-of-the-art semi-automatic segmentation framework [52]. The advantage of this algorithm is that it is very good at propagating segmentation information within a volume. However, it fails almost completely when propagating segmentation information across volumes. We use this segmentation algorithm to create dense pseudo-labels for the remaining unlabeled voxels $x_{l+1}, x_{l+2}, \ldots, x_{l+u}$.

Second, we use a more powerful DL-based segmentation algorithm to learn from the pseudo-labels of the weak segmentation algorithm. The advantage of this algorithm is that it can learn how to generalize across different volumes. However, it is significantly more data-hungry, and it is difficult to train it on the sparse user-provided labels only. The challenge in our context was to adapt existing deep learning architectures to our data and tasks. After training, the segmentation network can predict class probabilities for each trained class, summing up to 1.0, for each voxel. Using this semi-supervised two-stage labeling approach, where manual labels are first propagated to subsets of the data—which is next used for training of a more general segmentation problem—also reflects other machine learning concepts, such as self-training [51], [53] and distillation [54]. In the following subsections, we showcase the unique difficulties of cryo-ET data. Then we provide more details about each of the two segmentation algorithms. Finally, we discuss data management strategies for avoiding memory issues during training and the training protocol.

4.1 Data

Our dataset consisted of 60 cryo-ET volumes containing several SARS-CoV-2 virions each. One volume from the used dataset is available and has been deposited in the Electron Microscopy Data Bank[2] under id EMD-33297.

During the cryo-ET data acquisition process, the electrons must be very carefully spread throughout the whole tilt series in order to avoid damaging the specimen. This is conveyed in a single slice of the volume, where individual structures can be recognized but the objects of interest are mostly masked by noise. An example of one slice of such data is shown on the left side of Figure 3, where several SARS-CoV-2 virions are present. While some of them are just perceivable to the naked eye, others easily blend with the surrounding noise. This make images hard to segment not only for the untrained eye, but also for neural networks.

Each Cryo-ET volume has a resolution of $1024 \times 1440 \times [227 - 500]$ voxels. The raw data is stored with 32-bit precision, resulting in 122 GB for all volumes. During training, this data is converted to 16-bit precision due to mixed-precision training. The pseudo-label data can be stored with 16-bit precision, resulting in 61.3 GB per class. Following common best practices in deep learning, we split our 60 volumes into three sets: 50 volumes for training, five volumes for validation, and five volumes for an independent test set.

2. https://www.ebi.ac.uk/emdb/EMD-33297

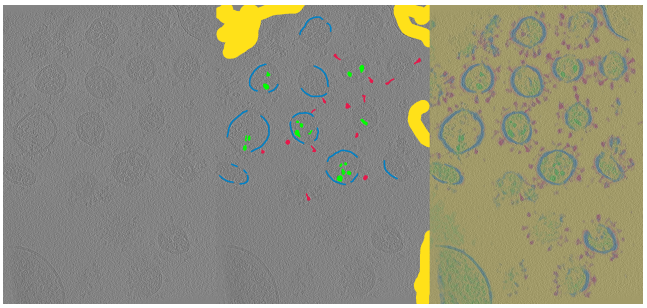

Fig. 3: Comparison of a single slice of the input data (left), sparse manual labels (middle), and dense pseudo-labels (right). Labels are background (yellow), membrane (blue), lumen (green), and spikes (red).

For this data, we use four classes corresponding to the *spikes*, *membrane*, *lumen* parts, and *background* of the SARS-CoV-2 virus cryo-ET data. To address a more general goal of *foreground-background* segmentation, we also experimented with two-class segmentation, which guided some decisions in the final model design.

4.2 Pseudo-Label Generation

The input to the pseudo-labeling stage is the raw cryo-ET volume. The outputs are either soft or hard segmentation pseudo-labels.

The pseudo-labeling of the data was completed using the Ilastik software [52]. We used a provided pixel classification pipeline with all 37 available 3D image features, covering intensity, edge, and texture properties, for propagating the sparse manual annotations throughout the volume. A comparison between the sparse manual annotations and dense pseudo-labels for a single slice is shown in Figure 3.

For each input volume, we defined four classes: (1) *Background*, (2) *Membrane*, (3) *Spikes*, and (4) *Lumen*. In our foreground-background segmentation, we only use the *Background* class and combine the other three classes into a *Foreground* class. On average, an experienced image-segmentation user spends around 30 minutes creating sparse manual labels for a single volume. The amount of manual annotations and the number of annotated slices along each axis depends on the individual volume and its content. After the manual labeling, an additional 1.5-4 hours of computation time is needed to propagate the labeled features to the whole volume and produce the pseudo-labels, which are validated by the annotators (see Figure 3 for a single slice of annotations). Ilastik works well for single-volume segmentation and propagating sparse user annotations to the whole volume, but not for propagating labeled features from one volume to other volumes (see Figure 4) and for this reason all the volumes were sparsely labeled manually. In this study, we use Ilastik as is and do not investigate the reasons why it fails to propagate the labeled features to the other volumes, contrary to one of its intended use cases on 2D images. The segmentation algorithm in Ilastik cannot separate structure from noise. By contrast, our proposed combination of two learning algorithms, drawing from semi-supervised learning, produces far better results, as demonstrated in the results section of this paper. The pseudo-

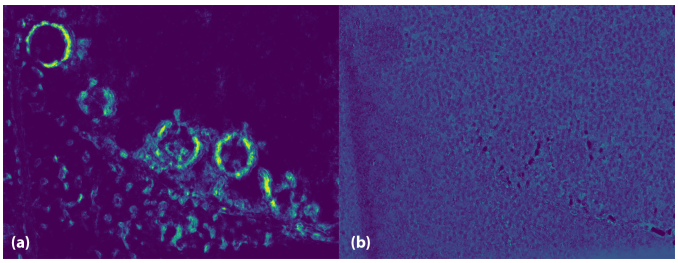

Fig. 4: Comparison of a segmentation slice of the same volume using labeling parameters trained on manual annotations of the volume with the use of labeling parameters estimated either (a) on another volume or (b) in Ilastik.

label generation stage can optionally provide soft or hard labels as output.

4.3 Manual-Label Generation

To further support the decision of using pseudo-labeled data as training data in our system, we manually labeled a sub-volume of the data from the test set. The segmentation was performed manually on a per-voxel basis by an experienced annotator and was validated by the domain expert. Due to the highly time-consuming nature of manual annotation of such noisy data, we segmented a single SARS-CoV-2 virion within a sub-volume of $256 \times 256 \times 256$ voxels. The annotation of four classes: *Background*, *Membrane*, *Spikes*, and *Lumen* took 56 hours and was performed using Amira Avizo[3].

4.4 Neural Network Architecture

To make well-informed design choices, we compared three different neural network architectures for foreground-background segmentation. The number of 3D segmentation approaches has grown rapidly in recent times, however, most approaches do not address the TEM modality; therefore, it is unclear exactly what constitutes the state-of-the-art approach for our segmentation problem. To counter this, we chose representatives of three major branches that have emerged from prior work on 3D segmentation, namely the standard U-Net, a residual learning approach, and a DenseNet. This choice accords with the findings of Moebel *et al.* [33], the most closely related to our approach, where authors used a relatively standard 3D U-Net in their recent work on macromolecule localization. Based on the experimental results using the respective official implementations, we selected the most suitable model taking into account its performance and training duration.

The first architecture is the 3D U-Net [21]. It contains an encoder (contracting path) and a decoder (expanding path). The encoder part extracts features from the studied volume, and each of its layers contains two $3 \times 3 \times 3$ convolutions, followed by a rectified linear unit (ReLU). The decoder upsamples the compressed volume back to its original resolution. Additionally, there are skip connections between the corresponding encoder and decoder layers where the resolution matches. This provides high-resolution low-level features to the decoder.

3. https://www.fei.com/software/avizo3d/%C2%A0

TABLE 1: Details of three networks

NETWORK	LAYERS	PARAMETERS (M: MILLIONS)
3D U-Net	18	19M
3D U-Net+ResNet	27	1.5M
3D DenseNet	47	1.6M

The second architecture is the 3D residual symmetric U-Net [24] (3D U-Net+ResNet). It contains U-Net's three main components (encoder, decoder, and same-scale skip connections). To enhance the propagation of volumetric context information, each layer is set up as a residual sub-network instead of using standard convolution layers. The worst reconstruction quality along the z-axis should be taken into account by omitting downsampling along z, as well as by using anisotropic convolution kernels ($7 \times 7 \times 5$) to match the anisotropic nature of the reconstructed volume data.

The last architecture is the skip-connected 3D DenseNet [27] (3D DenseNet). This network also includes a contracting and expanding path. To increase the receptive field of the feature maps, the contracting path contains four dense blocks. Each dense block contains four layers consisting of $3 \times 3 \times 3$ convolution, batch normalization, and ReLU activation with growth rate $k = 16$. There are direct connections from every layer to all subsequent layers. These connections help strengthen feature propagation. To utilize the multiple scale features in the intermediate dense blocks, the expanding path contains four 3D-upsampling operators to directly upsample the low-resolution features to the output resolution. The properties of each network are presented in Table 1.

Using the official implementations, we experimentally determined that all the network architectures are suitable for use in such cases but that the standard 3D U-Net approach is on par with 3D U-Net+ResNet and outperforms 3D DenseNet. Moreover, the training of 3D U-Net-based networks is far shorter and the U-Net architecture is much simpler than DenseNet. This led us to choose the 3D U-Net variant for subsequent experiments (see subsection 6.1).

To specialize the model and to retain generality for a background extraction that can be applied to other data in the cryo-ET domain, we trained the forward-background model with soft pseudo-labels. The soft segmentation aligned well with the visualization task, retaining the *soft information* throughout the pipeline. For further specialization, as demonstrated on our data, we retrained the model for four-class segmentation, this time with hard labels to maximize the distance between classes. We only needed to retrain the final layer of the model, which is much faster than training the model from the beginning. Moreover, the results were comparable to direct four-class soft and hard segmentation, as can be seen in Figure 18 in the Supplementary Material. Not only was the training faster (4 instead of 5 hours per epoch), it is also easy to change the specialization by only retraining the last layer. To cope with soft labels in the foreground-background segmentation, we considered this segmentation as a regression task, where predictions represent the probability of a voxel belonging to a specific class. We experimented with different activation functions for the output layer of the network. We tested *sigmoid*, *softmax*,

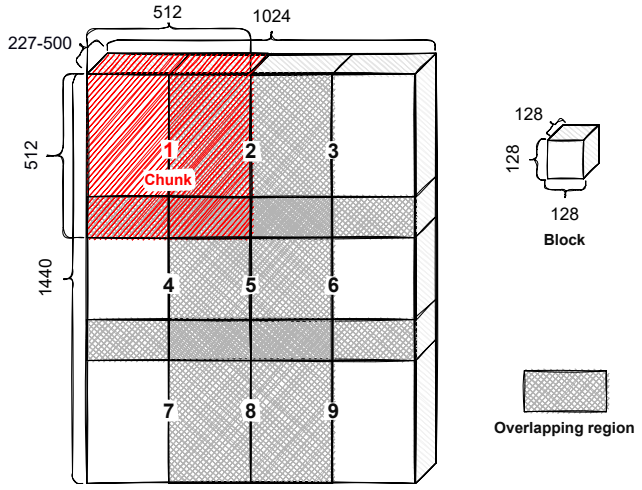

Fig. 5: Schematic of an individual cryo-ET volume divided into nine partially overlapping chunks.

normalized ReLU [34], and without activation, and paired them with appropriate loss functions, such as binary cross-entropy (BCE), mean squared error (MSE), and adaptive wing loss (AWL) [55]. AWL was initially proposed for heatmap regression, where regions with high intensity are usually more relevant for accurate predictions, while a low-intensity background can be very blurry. In practice, use of this loss tends towards the MSE on the background prediction and towards the mean absolute error on foreground predictions. We included this loss in our experiments because the described loss behavior is also desirable in our soft segmentation.

4.5 Data Management

Due to the high resolution of our volume data, it is currently infeasible to train deep models on whole volumes directly, as we run into memory limitations. To alleviate this issue, we train our models only on blocks of size $128 \times 128 \times 128$ voxels. Since our aim is to train on the blocks in a random order to prevent catastrophic forgetting, we must load the blocks from multiple volumes simultaneously, which introduces a storage bottleneck. To deal with this bottleneck, we divide each of our volumes into nine partially overlapping chunks of size $512 \times 512 \times [227 - 500]$ prior to blocking to reduce their storage footprint; see Figure 5. The chunks significantly overlap to avoid artifacts at the chunk borders. We also applied min-max normalization for the whole volume before splitting it into chunks. We iterate over all the chunks in each epoch during the training and randomly crop the $128 \times 128 \times 128$ input block from each chunk. During the inference, we block the full volume into the 128^3 blocks with an overlap of $32 \times 32 \times 32$ voxels. To get the prediction of a whole volume from overlapping blocks, we perform alpha-blended stitching in the overlapping regions.

4.6 Training

We used PyTorch [56] to implement the network architectures. For two-class segmentation, we experiment with both soft and hard labels as targets. For hard labels, we train each network with binary cross-entropy loss. The performance of the three network architectures is reported in Table 2 and discussed in subsection 6.1. We select the two best performing networks (3D U-Net and 3D residual symmetric U-Net) for evaluation with soft labels using MSE and AWL. The evaluation shows that the 3D residual symmetric U-Net with MSE loss performs best for such a task.

Optionally, we experiment with pre-training the four-class segmentation network with weights from the two-class segmentation network. In this case, we take the learned weights from the best performing two-class model, omit the last layer, and transfer them to a new model used for four classes. The last layer of the new model is adapted to output four probabilities and is then fine-tuned for the four-class segmentation. This is achieved by changing the number of output channels of the last layer from one to four, initializing this layer using random weights, and retraining the model. We use cross-entropy loss in this step. Note that our network still outputs soft labels (continuously-valued probabilities) that we use to visualize the segmentation certainty to the user, regardless of it being trained with hard or soft labels. The weights of the network are optimized using the Adam optimizer [57] ($\beta_1 = 0.9, \beta_2 = 0.999$) with a batch size of 4. We use a learning rate of 0.001 and weight decay of 0.0001 for regularization and mixed-precision training [58] to alleviate memory limitations.

5 Visualization Pipeline

The visualization pipeline leverages the neural network trained in the model generation stage to obtain the probabilistic segmentations of new volumes needed in the TF estimation and rendering stages described below.

5.1 Opacity Transfer Function Estimation

In our visualization system, we combine soft segmentation and raw input data. While we could attempt to extract some local geometric features from the raw data, we cannot do the same for the segmentation. The segmentation is obtained with deep neural networks from pseudo-labels, and we cannot assume that the local geometric features predicted by such a model reflect the corresponding geometric features in the original raw data. For this reason, we do not want to rely on any complex TF design process that takes into account such features and/or gradient information. We follow the simple ramping approach, which proves to produce good results while not relying on complex properties. In addition to that, the simplicity of the ramp TF allows simple modifications and fine-tuning by domain experts after the initial TF configuration is estimated automatically. After manual experimentation, we realize that the right limit of the ramp function should be at the end of the fuzzy value interval of $[0.0, 1.0]$, at 1.0 to obtain the best visual results, leaving only the left limit of the ramp as an unknown parameter that we will estimate. To find this ramp parameter, we first use a simple iterative image thresholding technique [59] outlined in Algorithm 1 for the calculation of an appropriate threshold value for each slice, the final volume threshold is a mean value of the calculated per-slice values. The algorithm iteratively extracts a foreground object from the background by determining the best threshold value

Algorithm 1: Algorithm for determining an appropriate threshold value of an individual slice. The final volume threshold is a mean value of the calculated per-slice values.

AutoThreshold $(slice)$
  Input : Slice - $slice$
  Output: Slice threshold - $sliceThreshold$

  $T = [], i = 0;$
  $imHist = \text{histogram}(slice);$
  $meanInt = \text{mean}(imHist);$
  $T[i] = \text{round}(meanInt);$
  do
    $meanIntBelowT = imHist < T[i];$
    $bgIntegrator = \text{mean}(meanIntBelowT);$
    $meanIntAboveT = imHist \geq T[i];$
    $fgIntegrator = \text{mean}(meanIntAboveT);$
    $i = i + 1;$
    $T[i] = \text{round}((bgIntegrator + fgIntegrator)/2);$
  while $(\text{abs}(T[i] - T[i-1]) \geq 1);$
  $sliceThreshold = \text{normalize}(T[i]);$

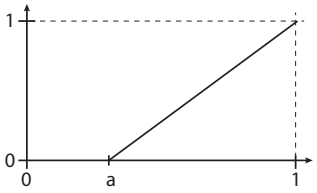

Fig. 6: Ramp TF. Value (a) is estimated for each class separately.

through frequency distribution analysis (for more details, see Supplementary Material). The thresholding algorithm is applied to all slices (in the vertical direction) of a segmented volume individually for each class i to take into account differences in volume due to the data acquisition (water-air boundary) and reconstruction (missing wedge). For final rendering, the threshold for an individual segmentation class i is calculated as:

$$a_i = \text{mean}\{\text{AutoThreshold}(slice_j); slice_j \in volume_i\}, \quad (1)$$

where $slice_j$ is j-th slice in the volume of the i-th class. The threshold value a_i is taken as the left limit of the ramp function r_i (see Figure 6) for the i-th class:

$$r_i(s) = \max\left(\frac{p_i(s) - a_i}{1.0 - a_i}, 0.0\right), \quad (2)$$

where s is a sample along the ray, $p_i(s)$ is the class probability for the i-th class at the sample location s in the corresponding volume, and a_i is the left limit of the ramping function. This could also be viewed as a separable 2D TF [43]. While there might be some additional benefits of using a non-separable 2D or even higher dimensional TF, we decide to keep the simple 1D TF design and minimize the user load.

Slice-based automatic thresholding did not return an equal value for all the slices of the volume. We found that the thresholds decreased for slices towards the top and bottom of the volume. We investigated the option of addressing this phenomenon in our TF—as the threshold changes can be nicely approximated with a simple quadratic function—but there is no significant improvement in the final visualization, making the use of a more complex TF less attractive.

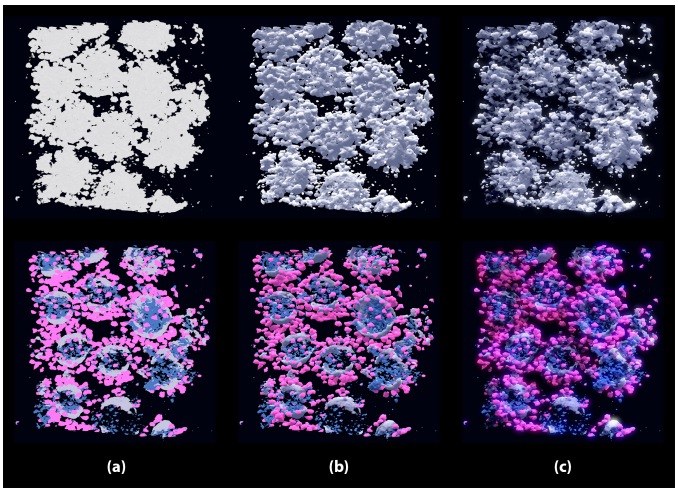

Fig. 7: Examples showing how the foreground-background data (top row) and four-class segmented data (bottom row) are rendered in different stages of the pipeline: (a) material color only, (b) added local ambient occlusion, and (c) added soft shadows and bloom. In all visualizations, the respective masks are multiplied by the low-pass filtered original data.

5.2 Rendering

Our goal is to achieve real-time speeds for the whole volume rendering, which limits the selection of the volumetric rendering technique. As an input, we take a raw cryo-ET volume V_{raw}, three soft segmentation volumes V_i, user-defined segmentation class colors c_i, and the estimated opacity TF parameters presented above, defined with a corresponding function r_i. The raw volume V_i is inverted and low-pass filtered (mean filter with 3-voxel radius) to emphasize the structures in the preprocessing step since the structures in the original data are represented with lower values.

In order to produce meaningful and clear visualizations of the fuzzy data, we avoid using normals for illumination. As normals would have to be estimated by calculating gradients, not only would we amplify the noise by using the normals, but we would also increase the amount of texture fetches that would have to be performed per sample. Instead, we approximate light scattering effects by sampling areas surrounding the illuminated voxels. We use a single spherical light to illuminate the scene for local shadow estimation. The rendering consists of four stages:

1) *Material color*: A unique user-defined base color is used for each segmentation mask, which must be significantly different from other classes to distinguish each class in visualization—see Figure 7 (a).
2) *Local ambient occlusion*: Local ambient occlusion (LAO) is calculated in the object space by randomly sampling a sphere around each voxel and calculating

the sum of the sampled voxel values from all the masks [60]. The number of samples can be defined by the user—for our test case, we used 50 samples. This adds local shadows that enhance the depth perception during the interaction. The sampling sphere is offset upwards so that the illumination of a surface is not attenuated by the voxels that lie below it. This in turn creates the illusion of shadows cast by a very large light source positioned above the volume. Results are shown in Figure 7 (b).

3) *Soft shadows*: Soft shadows (SS) are achieved using Monte Carlo integration of directional in-scattering, calculated in the object space by sampling a cone starting at a given voxel oriented towards a spherical light source [61]. The contributions of the other voxels are modulated by their distance to the original voxel. The SS help distinguish individual objects from each other and further enhance depth perception—see Figure 7 (c).
4) *Post-processing*: In the post-processing step, we add bloom by separating the brightest parts of the image via a tone mapping curve, blurring them with a Gaussian kernel, and adding the result to the original visualization. The purpose of the bloom is to increase the perceived dynamic range of the rendered scene by letting the brightest parts of the image contribute to the brightness of the surrounding areas. In combination with tone mapping, which can help reduce the clipping of the brightest image parts, the bloom effect can increase the perceived realism of the scene without obscuring any important details. This post-processing is executed in the screen space—see Figure 7 (c).

The first three steps of the rendering can be described by the following equations:

$$c_o = \sum_{s=s_0}^{s_{\max}} \sum_{i=i_0}^{i_{\max}} r_i(V_{i,s}) \cdot r_{\text{raw}}(V_{\text{raw},s}) \cdot c_i \cdot \text{ssc}_s \cdot \text{lao}_s \cdot a_{s,i}$$
$$a_{s,i} = \prod_{k=s_0}^{s} \prod_{j=i_0}^{i-1} (1 - r_j(V_{j,k}) \cdot r_{\text{raw}}(V_{\text{raw},k})) \quad (3)$$

where c_o is the output color, $s_0, \ldots, s_{max}$ is the ordered set of samples along the ray, $i_0, \ldots, i_{max}$ are the segmentation classes, $r_i(x)$ is the ramping function for the i-th class, $V_{i,s}$ is the s-th sample along the ray inside the i-th class volume, c_i is a user-defined material color for the i-th class, ssc_s is the SS contribution at sample s, and lao_s is the LAO contribution at the s-th sample. The second equation $a_{s,i}$ is the accumulated alpha on the ray up to sample s for class i.

The starting values of all the left ramp limits (for all volumes) are estimated with the approach presented in the previous section but are still user-adjustable.

The composition of the contributions along the rays cast through every pixel of the rendering canvas produces the real-time visualization demonstrated in the following section; a video is available in the online Supplementary Material.

6 Results

We tested our technique on a challenging but high-quality imaging dataset depicting SARS-CoV-2 virions. We were provided with 300 cryo-ET volumes of approximately 0.5 TB in total size. Due to limitations on the number of volumes that can be used for training and the availability of training segmentation, we limited our experimental dataset to 60 volumes. We performed a series of experiments on visualizing the noisy cryo-ET data and obtaining the fuzzy interpretations needed for visual mapping. We briefly summarize the outcome of these experiments below.

For the segmentation results, we begin by introducing the evaluation metrics used in our experiments before outlining our architecture selection, and lastly presenting the results of our best models for both soft foreground-background segmentation and segmentation into spikes, membranes, lumen, and background. For the evaluation of our visualization, we compare our technique with standard rendering techniques that are available in current off-the-shelf solutions and show a substantial improvement in visual quality.

TABLE 2: Performance on the validation set of three network architectures for binary segmentation

Method	BCE Loss	F1 Score	Train Time
3D U-Net	0.4280	0.6896	9h 10m
3D U-Net+ResNet	0.4360	0.6776	11h 58m
3D DenseNet	0.4701	0.6254	21h 21m

6.1 Segmentation Results

For evaluation, we use the F1 score or Dice similarity coefficient—a common evaluation metric for image segmentation—suitable for datasets that contain imbalanced class distribution. We define TP as the number of true-positive predictions, FP as the number of false-positive predictions, FN as the number of false-negative predictions, and TN as the number of true-negative predictions. The F1 score measures the similarity between the labels and predictions and is defined as

$$\text{F1} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}} \quad (4)$$

A higher F1 score indicates a better result. Note that the F1 score requires the use of discrete labels. To calculate an F1 score for our continuous values in the soft segmentation, we use a threshold of 0.5. This threshold is applied for both the label and prediction. In the four-class case, we use *argmax* to discretize the predictions. We consider pseudo-labels as the ground truth.

We evaluated different network architectures using binary labels, as detailed in subsection 4.6. We compared the standard 3D U-Net [21] with residual symmetric U-Net [24] (3D U-Net+ResNet) and 3D DenseNet [27] in terms of their predicted standard binary foreground-background labels on our data. Table 2 and Figure 15 in the Supplementary Material show the results of this experiment. The standard 3D U-Net and 3D U-Net+ResNet achieved a similar performance, clearly outperforming 3D DenseNet both in terms of F1 score and training time. Based on these results, we decided to further investigate the two 3D U-Net-based models.

In the next experiment, we trained the two 3D U-Net-based models for soft segmentation. In contrast to the

TABLE 3: Soft segmentation results comparing different activations (ACT) and loss functions.

MODEL	U-Net		U-Net+ResNet	
LOSS+ACT	LOSS	F1 SCORE	LOSS	F1 SCORE
MSE+NONE	0.0184	0.6567	0.0157	**0.6801**
AWL+NRELU	0.0409	0.5175	0.0561	0.3507

TABLE 4: F1 score comparison for manual and pseudo labels with our model predictions.

COMPARISON	F1 SCORE
Manual labels vs. 3D U-Net+ResNet	0.81
Pseudo labels vs. 3D U-Net+ResNet	0.77
Manual labels vs. Pseudo labels	0.70

TABLE 5: Rendering performance evaluation for both segmentations on two resolutions, given in milliseconds for each step of rendering: basic DVR, soft shadows (SS), and local ambient occlusion (LAO).

RESOLUTION	BASIC DVR	SS	LAO
FOREGROUND-BACKGROUND:			
Full-HD	16.59	2.55	12.14
4K	19.71	6.98	50.34
FOUR-CLASS:			
Full-HD	19.27	1.98	17.21
4K	35.01	6.08	46.93

binary label experiment, we faced a regression problem and, therefore, investigated several activation functions for the output layer and several loss functions. The results from this experiment can be seen in Table 3 and in Figures 16 and 17 in the Supplementary Material. The comparison shows that 3D U-Net+ResNet with MSE loss worked best. This is due to the decoder of this network, which outperformed the 3D U-Net. Also, note that the F1 score was comparatively low in this experiment. This was due to the fact that we needed to binarize both the soft labels and the predictions in order to compute the F1 score, which may have detrimentally affected numerically accurate predictions that were near the binarization threshold.

Lastly, we fine-tuned our best models from the previous experiment for use with all four classes. As our four-class labels are rather decisive, we fine-tuned the model using discrete labels instead of soft labels. We aimed to train a general foreground-background segmentation model for soft labels that could be used for other datasets in the same domain. In the case of SARS-CoV-2, there are four classes, so we fine-tuned the model for four-class labels. Another reason is that training directly on four classes would require too much time and too many resources. For foreground-background segmentation, we required 512 GB of RAM; with four GPUs, it took approximately 2 hours and 15 minutes per training epoch. In the case of four-class segmentation, because the size of data increases four times and memory limitation, the batch size had to be drastically smaller, further prolonging the training to taking 4 hours per epoch. The F1 score of this fine-tuned model on the validation set was 0.9773, and the cross-entropy loss was 0.0748. One of the reasons why the F1 score, in this case, was higher than in the foreground-background case is because of the nature of the F1 score. The F1 score is a good measure for incorrectly classified cases. In the foreground-background segmentation, there were two classes, so the F1 score penalized wrong predictions to a higher degree than in the four-class segmentation.

The training of the final foreground-background model took 5 days and 16 hours to converge. The fine-tuning of the selected transfer-learned model took an additional 2 days and 15 hours. The model inference took 20-25 seconds per volume on a single Nvidia V100 GPU computing node or 10-15 minutes per volume on a workstation with a single Nvidia Quadro RTX 8000 GPU.

The pseudo-labels were generated using Ilastik on a workstation computer with 2×Intel Xeon Gold 6230R @ 2.1 GHz, 256 GB of RAM, and a Nvidia Quadro RTX 8000 48 GB GPU running Microsoft Windows 10. The deep learning experiments were performed on diverse hardware: the model selection experiments were mostly performed on IBEX—a heterogeneous group of computing nodes—at KAUST, and the final model optimization was performed on a single computing node with 2×Intel Xeon Gold 6242 @ 2.8 GHz, 512 GB RAM, and 4×Nvidia Quadro RTX 8000 48 GB GPUs running Ubuntu Linux. The neural network's inference time was measured on both the machine used for labeling as well as the computing node.

For validation, we compared the performance of our model for manual and pseudo labels using the F1 score on an annotated subvolume. Additionally, we compared both types of labeled data, also using the F1 score. As can be seen from Table 4 they are all on par with each other. For the specific manually labeled subvolume, the model results are even closer than to the pseudo labels.

6.2 Rendering Results

The final visualizations of the proposed approach are displayed in several figures. The teaser image in Figure 1 shows how we get from the *solid* cryo-ET volume, over the foreground-background segmentation, to the four-class segmented rendering. We show segmentations and final renderings of a single virion segmented with the foreground-background model, as well as with the four-class model, in Figure 8. The dimension of this single virion sub-volume is $246 \times 264 \times 340$ voxels.

For visualization purposes, the visualization input data—soft segmentation generated with our model—was reduced to 8-bit precision. In the case of the foreground-background segmentation visualization, two volumes were loaded to the GPU, consuming 0.69 - 1.37 GB of GPU memory. In the case of the four-class segmentation visualization, four volumes were loaded to the GPU, resulting in 1.37 - 2.75 GB of memory use. We also tested our system on the full 16-bit precision—resulting in 2.75 - 5.49 GB of memory consumption. Apart from the loading times, there were no other differences in the performance, *i.e.*, the rendering times were the same. This shows that the method is compute and/or texture fetch bound and not bandwidth bound. In Table 5, we see how rendering performance varies by segmentation method and viewport resolutions.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3186146

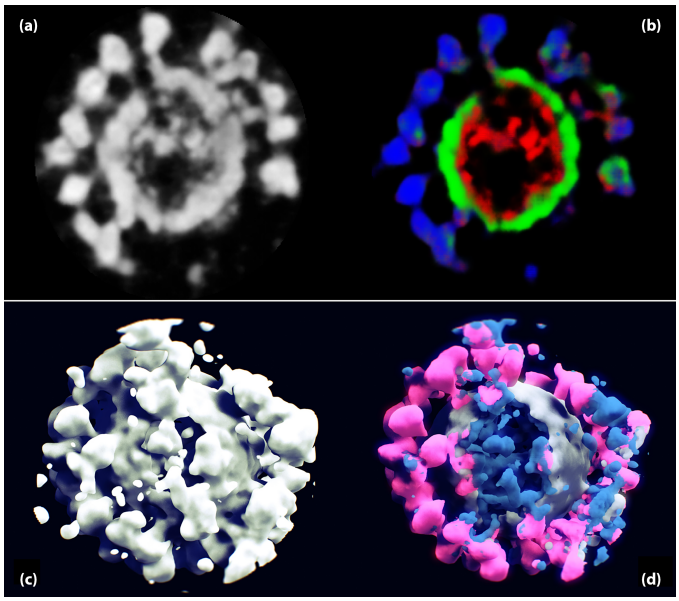

Fig. 8: One slice comparison of a single virion segmented with the foreground-background approach (a) and with the four-class approach (b)—spikes in blue, membrane in green, and lumen in red. The 3D visualization of these segmentations with our rendering pipeline is shown in (c) and (d), respectively—spikes in pink, membrane in gray, and lumen in blue.

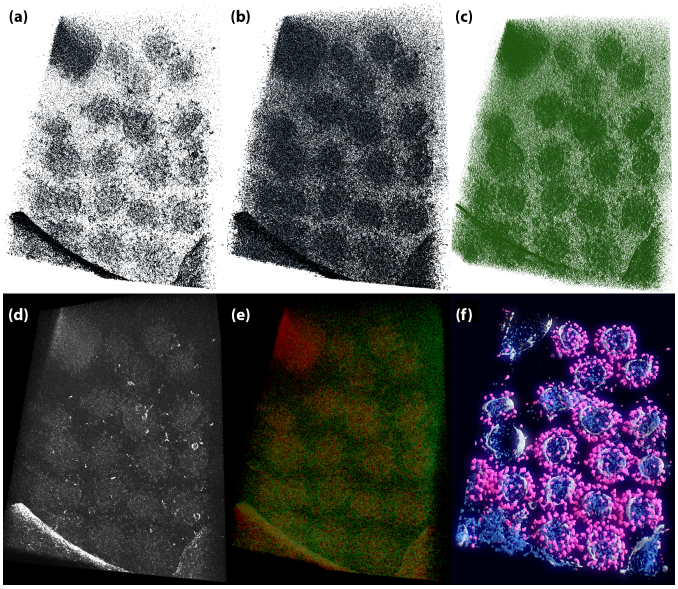

Fig. 9: Comparison with other DVR techniques: (a) and (b) ISO surface rendering with two different ISO values, (c) VPT, (d) MIP, (e) EAM with color TF, and (f) our approach.

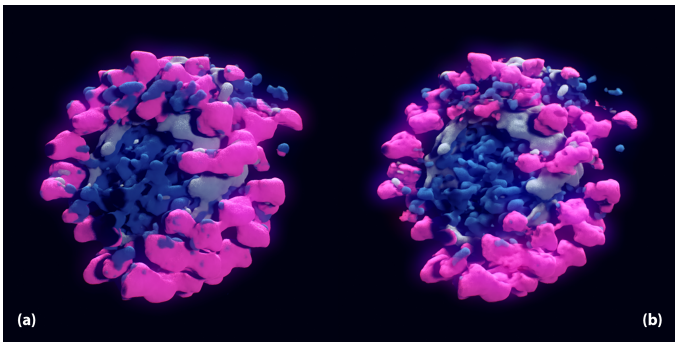

Fig. 10: Comparison of our method using only the segmentation volume (a), and multiplying the segmentation volume with the raw data (b).

We ran an automated visualization task of 360° tilt rotation of all five test volumes for the foreground-background segmentation (two volumes) and the four-class segmentation (four volumes). Each experiment was run five times on full-HD (1080p) and 4k (2160p) resolution, respectively; then, the measurements were averaged. The portion of the screen covered by the volume rendering was the same for both resolutions. The visualization evaluation was done on a workstation computer with 2×Intel Xeon Gold 6230R @ 2.1 GHz, 256 GB of RAM, and an Nvidia Quadro RTX 8000 48 GB GPU running Microsoft Windows 10.

To show the difficulty of rendering the cryo-ET data directly, we showcase comparisons with other volume rendering approaches of the raw data: ISO surface rendering, emission-absorption model (EAM), maximum intensity projection (MIP), and volumetric path tracing (VPT). All these methods were used for rendering on a 2k canvas and run in real-time, except VPT, where one needs to wait for the convergence. Since the results using the original data were unusable, we show the renderings performed on inverted low-pass filtered data, which we also used in our approach to suppress high-frequency noise and emphasize the specimen structures. The visual comparison with other DVR techniques is displayed in Figure 9. The output images might significantly vary with the use of different rendering parameters, *e.g.*, ISO values and transfer functions. In the presented examples we used two different ISO-values and tried to set the best 2D transfer functions we could. While the basic shape outlines are visible—even with inverted low-pass filtered data—the details are unrecognizable. There is also no easy way to configure a 2D TF to distinguish among the four classes we segmented in our approach. In Figure 10, we show the impact of including the original data in the visualization. Specifically, we multiplied the original data with the segmentation volumes after opacity mapping of each volume (with low-pass filtered data) according to the equations 3 to reveal the fine details and structures of the original data.

To further demonstrate the difference between our approach and common DVR techniques, we used the data from our preprocessing pipeline for rendering with common DVR techniques with the same parameters. In Figure 11, we see that most details were retained with our approach (a), while EAM (b) blurred the structures, ISO (c) failed to show all the desired data, and MIP (d) propagated values from structures in the back to the rendering. For comparison, we also added the VPT (d) results obtained after 30 minutes of convergence, which showed the most details. The top row shows the rendering without LAO and SS; the bottom row adds both (except for VPT).

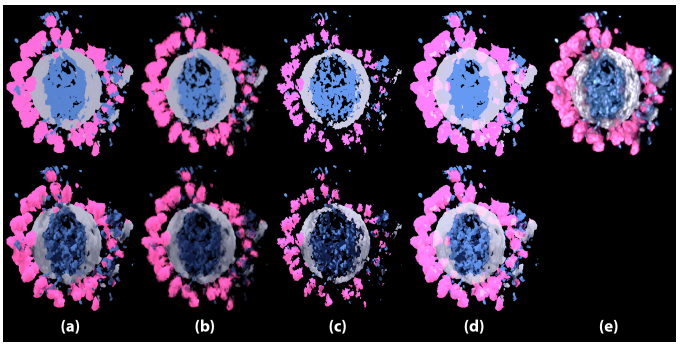

Fig. 11: Comparison of our approach with other DVR techniques. The top row shows the output of regular DVR approaches, while in the bottom row we add LAO and SS: our approach (a), EAM (b), ISO (c), MIP (d). Image (e) shows VPT after 30 min convergence.

7 Discussion

A common problem in biological research, especially in cryo-EM, is the lack of ground-truth data. One of the hazards of basing a visualization technique on segmented data is that it is difficult or impossible for a researcher to detect errors in the segmentation pipeline. Traditional approaches used for obtaining such visualization consist of manually segmenting the data or combining denoising and thresholding steps. The denoising is achieved with low-pass filtering or DL-based denoising methods. While complete manual segmentation is not feasible for large datasets, we show how our approach compares to the denoising and thresholding approach. In Figure 12, we present the results on a single slice of a tomogram. We compare the thresholded results after applying the mean-3 filter and Topaz Denoise [13] method to the data with our approach. Specifically, we compare local and global automatic thresholding methods implemented in the Fiji [62] image processing package. The in-depth comparison of automatic thresholding is presented in the supplementary material. We manually selected the results of the best performing method, which produced results with the least visible noise and highest specimen preservation. For mean-3 filtered data, this was Yen [63] global thresholding method, and Moments [64] global thresholding method for Topaz Denoise denoised data. Still, this is only foreground-background separation. Further segmentation is traditionally performed using a DL approach which is in line with our proposed approach.

We discussed the results of our work with two domain experts. One domain expert is the co-author of this paper, and the other is an independent domain expert.

The first expert is a physicist specializing in biophysics. He has 12 years of experience in the field and eight years in cryo-ET. He is head of the cryo-ET laboratory at his university and works with cryo-ET data almost daily, of biological specimens only. He was involved in the design process and provided the motivation from the view of a structural virologist. He was in charge of the acquisition of the data used in this study.

He explained that experts mostly preprocess data with a Gaussian filter and analyze it manually (*e.g.*, selecting each spike position manually on 2D slices). They perform weeks-long reconstructions of structures before they are able to see them in 3D. He confirmed that our foreground-background pseudo-labels are good but could be improved with voxel-precise manual annotation, especially for the lumen structure annotations, where specialized domain knowledge would be beneficial. Moreover, he suggested that creating pseudo-labels might be easier on preprocessed (inverted and low-pass filtered) data, which should be investigated in the future. He confirmed that the results are satisfactory for the amount of time spent on the segmentation task and that full voxel-precise manual annotation, which is used for the parts of the volumes in their pipeline, would take up to several days per volume. He suggested omitting the top-most and bottom-most parts of the volumes where there are artifacts due to an air-water boundary.

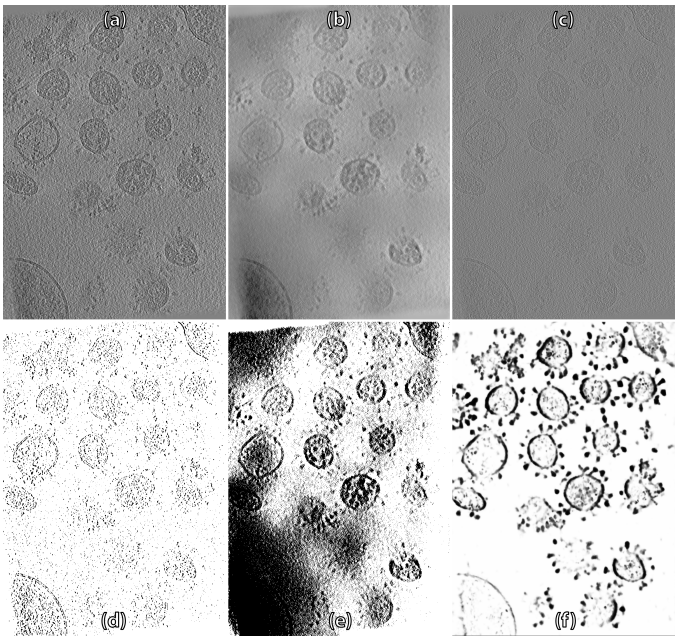

Fig. 12: The comparison between traditional foreground-background segmentation techniques: (a) mean-3 filtered data and (d) thresholded results using Yen [63] (e); (b) denoised data using the Topaz Denoise method and (e) thresholded results using Moments [64]; (c) original data and the background classification using our approach.

He also confirmed that our neural network's predictions are good. Moreover, he supported that, in some cases, the results are even better than the pseudo-labels, exposing several structures that were previously not identified as well as making the class borders sharper. Such structures are presented in Figure 13. He suggested that a *dust removal* approach could further improve the segmentation by removing particles smaller than a given diameter.

He also validated our labeling of the data. After reviewing the four-class pseudo-labels, he concluded that the labeling was performed well but could be further improved with voxel-precise segmentation. This is true for the lumen structures and the portions of the membrane and spike annotations (see Figure 14 (a), (c), and (e)). It was apparent that there were still some spike outlines in parts of the membrane and vice versa. He suggested addressing this in the future by trying to impose some local limitations

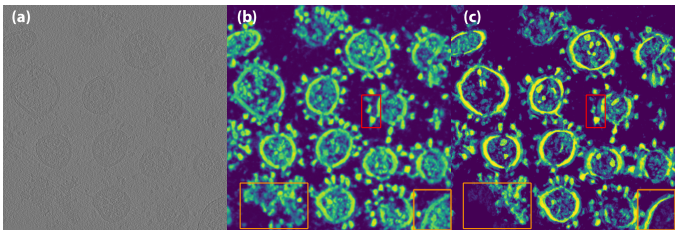

Fig. 13: Comparison of original input data (a), data segmented with our model (b), and pseudo-labels (c). In the orange boxes parts of viruses that were not labeled during the pseudo-labeling process but were segmented with our approach are shown. In the red boxes the areas between the membrane and spikes, which are cleaner in the results from our model, are shown.

to the pseudo-labeling process. After seeing the automatic segmentation results, he was positively surprised at how well the membrane and spike segmentation performed (see Figure 14 (d) and (f)). He was delighted that the spikes were also present in the *missing wedge* regions, where he did not expect such good results (see the right-most virions in Figure 1 and Figure 8 (d)). He confirms that the lumen segmentation could be further improved, possibly with better pseudo-labels. Furthermore, under the expert's supervision, we voxel-precisely labeled a subvolume of the dataset and he validated our labels as being as good as if they would be produced by the domain expert.

We introduced him to three visualization approaches: EAM, MIP, and our approach. He confirmed that he is familiar with the basic DVR visualization techniques and that in his work he mostly uses the visualization pipeline integrated into Chimera [65]. He confirmed that basic DVR methods are not suitable for direct rendering of the data and that our approach is excellent. He supported the claim that the details added from the original tomogram data enhanced the surface structure's comprehension. He pointed out that existing visualization packages have additional functionality, which is very useful to the researchers—such as *dust removal*—but is orthogonal to the available visualization settings. He supported that manual transfer function adjustment is beneficial for fine-tuning the output. Furthermore, he suggested adding options to change the lighting conditions, which would help explore further details of the specimen. Additionally, we asked the expert how the output of different rendering stages helps with the perception of structures in the data. He said that adding local ambient occlusion helps the most with shape comprehension and depth perception. Soft shadows and post-processing are less beneficial but still help in distinguishing smaller structures apart from one another.

The second expert is a cell biologist specializing in electron microscopy. He has more than 19 years of experience in EM and 15 years with Transmission Electron Microscopy specifically. He is a team lead of the electron microscopy laboratory at our university and works with EM/ET data on average three times a month. At first, his work only included biological specimens, but in the last 12 years he has also worked with polymer membranes and catalytic nanoparticles. We have not collaborated with this domain expert, and we regard his feedback as fully independent.

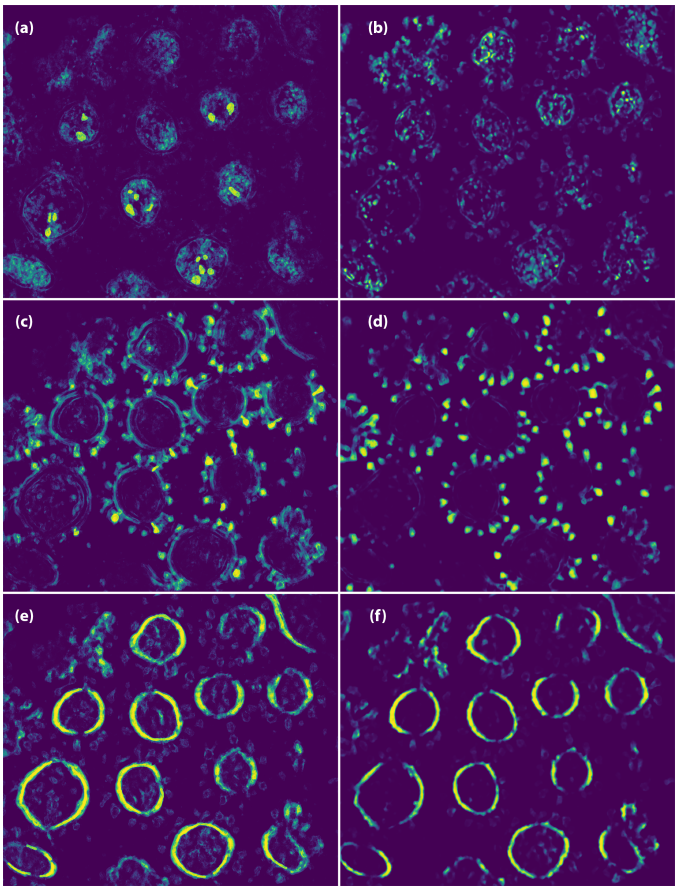

Fig. 14: Comparison between four-class segmentation of membranes (bottom), spikes (middle), and lumen (top) for pseudo-labels (left) and results of our model (right). The clearer output produced by our model is visible. The background class is not displayed.

His first impression was that the data are good. They are well aligned, the missing wedge is apparent and supports the good alignment, and the fiducial markers were mostly removed. He also confirmed that the pseudo-labels are good, especially for the time spent per volume since good fully manual annotation could take up to several days per individual volume. Due to the resolution of the data, the segmentation could be further improved by separately selecting individual layers of the lipid bilayer membrane.

He also confirmed that foreground-background segmentation by our model is good. To some extent, it gave even better insight into the structures than pseudo-labels. While the spikes were sharper in the pseudo-labels, they sometimes overlapped and merged. On the other hand, the space between the spikes and membrane were often smudged in the pseudo-labels; the space was much *cleaner* in the results from our model (see red boxes in Figure 13). Visually, the results from our model were closer to the real data due to less saturated areas. He also pointed out that, in many occurrences, our model found more structures than the pseudo-labeling process (see orange boxes in Figure 13, and the value bleeding was more prominent in the pseudo-labels than in the results from our model, which showed

cleaner structures.

We showed him three methods of 3D visualization of the data: EAM, MIP, and our approach. He pointed out that he is familiar with even more visualization techniques available in the EM-related software (IMOD [66], Amira Avizo, and Chimera [65]). He furthermore mentioned that common volume rendering techniques (MIP, ISO) are not suitable for visualizing such data. He mostly uses transparency-based methods (EAM) that give better insight into the underlying data structure. Our visualization approach was very appealing to him. It presented the structures as solid while still maintaining the surfaces' textural details that reveal the specimen's detailed structure.

Finally, he pointed out that automation in the segmentation and visualization of EM data is crucial for accelerating the field's development. As it was pointed out by the expert, our work addresses this cryo-ET bottleneck and enables scientists from these domains to drastically speed up their research. We present one quote from his statements during the interview: "*I doubt that you are fully aware about how impactful your technology will be for our field, once it becomes available as a tool.*" Our method allows for a quick inspection of the acquired data, making it easier to prioritize which data should be further analyzed first. In the past, vast datasets were acquired but never selected for analysis. With the use of the presented approach, one could revisit these datasets to discover potential interesting specimens.

The experts agreed that *"From the view of a structural virologist, the method proposed in this work offers a quick, semi-automatic way to quickly de-noise the cryo-ET data. The result looks far better than simply applying an adapted Gaussian filter. From the results, at least the number of spikes per virus is easily discernible, which is very helpful for both vaccine developers and structural biology work."*

Even though we used generated pseudo-labels for training our deep model, we showed that this is a viable alternative to the use of manually labeled training data only. The same was also shown in [33], where authors compared two ways of preparing the annotations for training the deep model: (1) an alignment of subtomogram averaged structure shape or (2) a sphere with a user-defined radius for individual structure type. This saves an enormous amount of laborious manual work for the field experts. Full manual annotation of datasets of the same size would take several orders of magnitude more time than pseudo-labeling. Moreover, the expert supported that manual labeling is very rarely still used in the field, not only because of its time demand but also due to the subjective nature of annotations. Not only do the segmentation distinguish between the individual annotators by a margin, but they also differ a lot for the same annotator throughout the annotation process. By using pseudo-labels, we largely omit these problems.

The second expert also provided us with feedback on the impact of different rendering stages on the final visualization. While the basic DVR gives a good insight into how the data is structured and where the individual classes are, the expert confirmed that LAO adds much-needed shading of the structures, which enables users to easier comprehend their shape and depth relation. Additionally, he expressed that this makes the LAO stage very important for comprehending structures in the volume. Moreover, he agreed that the SS and post-processing stage with tone mapping and bloom are also beneficial, but to a lesser extent. From his observations, he concluded that the SS stage enables users to more easily distinguish between individual structures close together, and the post-processing stage, which includes tone mapping and bloom, additionally emphasizes the fine details of the structures.

As with all the DL-based segmentation approaches, also the presented approach contains uncertainty. By interacting with the lower ramp TF parameter (a in Figure 6) for an individual class, the users can obtain insight into the uncertainty of the particular segmentation class.

A number of vaccines are based on inactivated and purified viruses. However, SARS-CoV-2 spikes are known to be fragile and can easily detach or shed their S1 subunits when subjected to physical shock, a freeze-thaw process, or improper chemical inactivation. A recent study [67] showed that 74% of spikes on the BPL-fixed SARS-CoV-2 viruses were damaged. Such fragility imposes challenges for vaccine development since losing or damaging spikes means weakened antigenicity or a less effective immune response. Therefore, cryo-ET offers a high-resolution control for checking the intactness of the antigen number and structure of the inactivated virus. The method proposed in this work offers a quick and reliable way to count the number of antigens.

For structural biology, our proposed method offers a quick way to segment and identify the spikes of SARS-CoV-2. Once their positions are converted to 3D coordinates, the spikes can be cropped out for subtomogram averaging and solving structures. Prior to using this method, which will save a lot of time, cryo-ET specialists typically established the coordinates through tedious manual identification or unreliable template-matching.

The overall feedback from the experts was very positive. They confirmed that meaningful and expressive visualization is crucial for understanding the data. They gave us some directions for possible future extensions of the presented work in the segmentation and the visualization aspects. They agreed that having such a tool in their processing pipeline would definitely speed up and simplify their analysis workflow.

8 Conclusion

With this work, we have shown how, with a suitable approach and a set of tailored techniques, one can *excavate* and visualize information hidden by an extremely low signal-to-noise ratio, such as within cryo-ET data. We demonstrate how the power of deep neural networks can be harnessed to infuse the visualization pipeline with detailed automatic segmentation, yielding high-quality visualization results, where common volume rendering approaches using a simplistic TF fail.

There are limitations to using such a system for a specific pipeline by feeding it with specific segmentation in the training step. However, it still promises to save days if not weeks of laborious manual segmentation work for visualization specialists while achieving great visual results. By providing even more precise segmentation data—instead of fuzzy semi-automatic segmentation—to our system, the

final visualization is expected to be even better. Embedding our system into the cryo-ET pipeline would give domain scientists access to high-quality data visualization with almost no additional effort. The segmentation that domain experts prepare daily can be used as training input to our system. Moreover, using the data from different laboratories working on similar problems can lead to the development of specialized visualization models for specific cases, which could be shared and benefit the research community.

While we tested our method on data obtained using the same sample preparation and data acquisition pipeline, it is still necessary to investigate the degree to which the parameters in the preparation and acquisition process can vary for such an approach to still work well.

While the inference part of the system is fast—20-25 seconds per volume—it is still not interactive. This could be achieved if the inference was performed per block/chunk, which would also allow for a fast change of the model used in the visualization pipeline.

The presented system could be further extended to become an end-to-end deep learning system. Not only fuzzy segmentation masks, but other visualization parameters (*e.g.*, TF parameters and rendering parameters) could also be trained for a specific domain. This would include a differentiable volumetric rendering system, which would allow the optimization of rendering parameters.

Our next step will be to integrate the volume visualization into data preparation pipelines for the subtomogram averaging process. The results of this signal-to-noise ratio amplification methodology could be fed back into our volume visualization pipeline to further enhance details and potentially address the missing wedge artifact.

Acknowledgments

The research was supported by King Abdullah University of Science and Technology (KAUST) (BAS/1/1680-01-01), the KAUST Visualization Core Lab, the Baden-Württemberg Stiftung through the ABEM Project, and in part from Tsinghua University Spring Breeze Fund #2021Z99CFZ004 (SL), National Natural Science Foundation of China #32171195 (SL). We thank nanographics.at for providing the Marion software. We would also like to thank Rachid Sougrat from the Imaging and Characterization KAUST Core Lab for the in-depth feedback on our results. Finally, we thank Jenny Booth, who works with the Research Communication team at KAUST, for proofreading, and anonymous reviewers for their constructive comments.

References

[1] W. Kühlbrandt, "The Resolution Revolution," *Science*, vol. 343, no. 6178, pp. 1443–1444, 2014. [Online]. Available: https://doi.org/10.1126/science.1251652

[2] K. E. Leigh, P. P. Navarro, S. Scaramuzza, W. Chen, Y. Zhang, D. Castaño-Díez, and M. Kudryashev, "Chapter 11 - Subtomogram averaging from cryo-electron tomograms," in *Three-Dimensional Electron Microscopy*, ser. Methods in Cell Biology, T. Müller-Reichert and G. Pigino, Eds. Academic Press, 2019, vol. 152, pp. 217–259. [Online]. Available: https://doi.org/10.1016/bs.mcb.2019.04.003

[3] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[4] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, K. Tunyasuvunakool, O. Ronneberger, R. Bates, A. Zidek, A. Bridgland *et al.*, "High accuracy protein structure prediction using deep learning," *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, vol. 22, p. 24, 2020.

[5] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume Rendering," *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 65–74, Jun. 1988. [Online]. Available: https://doi.org/10.1145/378456.378484

[6] R. I. Koning, A. J. Koster, and T. H. Sharp, "Advances in cryo-electron tomography for biology and medicine," *Annals of Anatomy - Anatomischer Anzeiger*, vol. 217, pp. 82–96, 2018. [Online]. Available: https://doi.org/10.1016/j.aanat.2018.02.004

[7] F. K. Schur, "Toward high-resolution in situ structural biology with cryo-electron tomography and subtomogram averaging," *Current Opinion in Structural Biology*, vol. 58, pp. 1–9, 2019. [Online]. Available: https://doi.org/10.1016/j.sbi.2019.03.018

[8] M. Chen, J. M. Bell, X. Shi, S. Y. Sun, Z. Wang, and S. J. Ludtke, "A complete data processing workflow for cryo-ET and subtomogram averaging," *Nature Methods*, vol. 16, no. 11, pp. 1161–1168, 2019.

[9] H. Yao, Y. Song, Y. Chen, N. Wu, J. Xu, C. Sun, J. Zhang, T. Weng, Z. Zhang, Z. Wu, L. Cheng, D. Shi, X. Lu, J. Lei, M. Crispin, Y. Shi, L. Li, and S. Li, "Molecular Architecture of the SARS-CoV-2 Virus," *Cell*, vol. 183, no. 3, pp. 730–738.e13, 2020. [Online]. Available: https://doi.org/10.1016/j.cell.2020.09.018

[10] X. Huang, S. Li, and S. Gao, "Exploring an optimal wavelet-based filter for cryo-ET imaging," *Scientific Reports*, vol. 8, no. 1, pp. 2582–2591, 2018. [Online]. Available: https://doi.org/10.1038/s41598-018-20945-6

[11] H. Shigematsu and F. Sigworth, "Noise models and cryo-EM drift correction with a direct-electron camera," *Ultramicroscopy*, vol. 131, pp. 61–69, 2013. [Online]. Available: https://doi.org/10.1016/j.ultramic.2013.04.001

[12] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," *arXiv preprint arXiv:1803.04189*, 2018.

[13] T. Bepler, K. Kelley, A. J. Noble, and B. Berger, "Topaz-Denoise: general deep denoising models for cryoEM and cryoET," *Nature Communications*, vol. 11, no. 1, p. 5208, Oct 2020. [Online]. Available: https://doi.org/10.1038/s41467-020-18952-1

[14] T. Buchholz, M. Jordan, G. Pigino, and F. Jug, "Cryo-CARE: Content-Aware Image Restoration for Cryo-Transmission Electron Microscopy Data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 502–506. [Online]. Available: https://doi.org/10.1109/ISBI.2019.8759519

[15] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2129–2137. [Online]. Available: https://doi.org/10.1109/CVPR.2019.00223

[16] M. Su, H. Zhang, K. Schawinski, C. Zhang, and M. A. Cianfrocco, "Generative adversarial networks as a tool to recover structural information from cryo-electron microscopy data," *BioRxiv*, p. 256792, 2018. [Online]. Available: https://doi.org/10.1101/256792

[17] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3059968

[18] C. Chen, C. Qin, H. Qiu, G. Tarroni, J. Duan, W. Bai, and D. Rueckert, "Deep learning for cardiac image segmentation: A review," *Frontiers in Cardiovascular Medicine*, vol. 7, p. 25, 2020. [Online]. Available: https://doi.org/10.3389/fcvm.2020.00025

[19] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017. [Online]. Available: https://doi.org/10.1016/j.asoc.2018.05.018

[20] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.11.118

[21] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Springer International Publishing, 2016, pp. 424–432. [Online]. Available: https://doi.org/10.1007/978-3-319-46723-8_49

[22] W. Zhu, Y. Huang, L. Zeng, X. Chen, Y. Liu, Z. Qian, N. Du, W. Fan, and X. Xie, "AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy," *Medical physics*, vol. 46, no. 2, pp. 576–589, 2019. [Online]. Available: https://doi.org/10.1002/mp.13300

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[24] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung, "Superhuman accuracy on the SNEMI3D connectomics challenge," *arXiv preprint arXiv:1706.00120*, 2017.

[25] T. D. Bui, J. Shin, and T. Moon, "3D densely convolutional networks for volumetric segmentation," *arXiv preprint arXiv:1709.03199*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1709.03199

[26] T. D. Bui, S.-i. Ahn, Y. Lee, and J. Shin, "A skip-connected 3d densenet networks with adversarial training for volumetric segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 378–384. [Online]. Available: https://doi.org/10.1007/978-3-030-11723-8_38

[27] T. D. Bui, J. Shin, and T. Moon, "Skip-connected 3D DenseNet for volumetric infant brain MRI segmentation," *Biomedical Signal Processing and Control*, vol. 54, p. 101613, 2019. [Online]. Available: https://doi.org/10.1016/j.bspc.2019.101613

[28] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Automatic 3D cardiovascular MR segmentation with densely-connected volumetric convnets," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2017, pp. 287–295. [Online]. Available: https://doi.org/10.1007/978-3-319-66185-8_33

[29] N. Siddique, P. Sidike, C. Elkin, and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3086020

[30] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv preprint arXiv:1804.03999*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1804.03999

[31] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11. [Online]. Available: https://doi.org/10.1007/978-3-030-00889-5_1

[32] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1055–1059. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9053405

[33] E. Moebel, A. Martinez-Sanchez, L. Lamm, R. D. Righetto, W. Wietrzynski, S. Albert, D. Larivière, E. Fourmentin, S. Pfeffer, J. Ortiz, W. Baumeister, T. Peng, B. D. Engel, and C. Kervrann, "Deep learning improves macromolecule identification in 3D cellular cryo-electron tomograms," *Nature Methods*, vol. 18, no. 11, pp. 1386–1394, 2021. [Online]. Available: https://doi.org/10.1038/s41592-021-01275-4

[34] C. Gros, A. Lemay, and J. Cohen-Adad, "SoftSeg: Advantages of soft versus binary training for image segmentation," *Medical image analysis*, vol. 71, p. 102038, 2021. [Online]. Available: https://doi.org/10.1016/j.media.2021.102038

[35] H. C. Cheng, A. Cardone, S. Jain, E. Krokos, K. Narayan, S. Subramaniam, and A. Varshney, "Deep-Learning-Assisted Volume Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 2, pp. 1378–1391, 2019. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2796085

[36] F. Hong, C. Liu, and X. Yuan, "DNN-VolVis: Interactive volume visualization supported by deep neural network," in *2019 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2019, pp. 282–291. [Online]. Available: https://doi.org/10.1109/PacificVis.2019.00041

[37] M. Berger, J. Li, and J. A. Levine, "A Generative Model for Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 4, pp. 1636–1650, 2019. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2816059

[38] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish, "A rule-based tool for assisting colormap selection," in *Proceedings Visualization '95*, 1995, pp. 118–125. [Online]. Available: https://doi.org/10.1109/VISUAL.1995.480803

[39] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *IEEE Symposium on Volume Visualization (Cat. No.989EX300)*, 1998, pp. 79–86. [Online]. Available: https://doi.org/10.1109/SVV.1998.729588

[40] C. D. Correa and K.-L. Ma, "Visibility Histograms and Visibility-Driven Transfer Functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 192–204, 2011. [Online]. Available: https://doi.org/10.1109/TVCG.2010.35

[41] S. Lindholm, P. Ljung, C. Lundström, A. Persson, and A. Ynnerman, "Spatial Conditioning of Transfer Functions Using Local Material Distributions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1301–1310, 2010. [Online]. Available: https://doi.org/10.1109/TVCG.2010.195

[42] L. Cai, W.-L. Tay, B. P. Nguyen, C.-K. Chui, and S.-H. Ong, "Automatic transfer function design for medical visualization using visibility distributions and projective color mapping," *Computerized Medical Imaging and Graphics*, vol. 37, no. 7, pp. 450–458, 2013. [Online]. Available: https://doi.org/10.1016/j.compmedimag.2013.08.008

[43] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the Art in Transfer Functions for Direct Volume Rendering," *Computer Graphics Forum*, vol. 35, no. 3, pp. 669–691, 2016. [Online]. Available: https://doi.org/10.1111/cgf.12934

[44] S. Luo and J. Dingliana, "Transfer Function Optimization Based on a Combined Model of Visibility and Saliency," in *Proceedings of the 33rd Spring Conference on Computer Graphics*, ser. SCCG '17. Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3154353.3154357

[45] B. Ma and A. Entezari, "Volumetric Feature-Based Classification and Visibility Analysis for Transfer Function Design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 12, pp. 3253–3267, 2018. [Online]. Available: https://doi.org/10.1109/TVCG.2017.2776935

[46] J.-S. Prassni, T. Ropinski, and K. Hinrichs, "Uncertainty-aware guided volume segmentation," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 6, pp. 1358–1365, 2010. [Online]. Available: https://doi.org/10.1109/TVCG.2010.208

[47] C. Lundstrom, P. Ljung, and A. Ynnerman, "Local histograms for design of transfer functions in direct volume rendering," *IEEE Transactions on visualization and computer graphics*, vol. 12, no. 6, pp. 1570–1579, 2006. [Online]. Available: https://doi.org/10.1109/TVCG.2006.100

[48] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman, "Uncertainty visualization in medical volume rendering using probabilistic animation," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1648–1655, 2007. [Online]. Available: https://doi.org/10.1109/TVCG.2007.70518

[49] S. Diepenbrock, J.-S. Prassni, F. Lindemann, H.-W. Bothe, and T. Ropinski, "2010 IEEE visualization contest winner: Interactive planning for brain tumor resections," *IEEE computer graphics and applications*, vol. 31, no. 5, pp. 6–13, 2011. [Online]. Available: https://doi.org/10.1109/MCG.2011.70

[50] D. hyun Lee, "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," 2013.

[51] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01070

[52] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk, "Ilastik: interactive machine learning for (bio)image analysis," *Nature Methods*, Sep. 2019. [Online]. Available: https://doi.org/10.1038/s41592-019-0582-9

[53] G. Bortsova, F. Dubost, L. Hogeweg, I. Katramados, and M. de Bruijne, "Semi-supervised Medical Image Segmentation via Learning Consistency Under Transformations," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Cham: Springer International Publishing, 2019, pp. 810–818. [Online]. Available: https://doi.org/10.1007/978-3-030-32226-7_90

[54] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a

Neural Network," *arXiv preprint arXiv:1503.02531*, 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1503.02531

[55] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6971–6981.

[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6980

[58] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. wu, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1710.03740

[59] T. Ridler, S. Calvard *et al.*, "Picture thresholding using an iterative selection method," *IEEE trans syst Man Cybern*, vol. 8, no. 8, pp. 630–632, 1978. [Online]. Available: https://doi.org/10.1109/TSMC.1978.4310039

[60] F. Hernell, P. Ljung, and A. Ynnerman, "Local Ambient Occlusion in Direct Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 4, pp. 548–559, Jul. 2010. [Online]. Available: https://doi.org/10.1109/TVCG.2009.45

[61] E. Veach, "Robust Monte Carlo Methods for Light Transport Simulation," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1997. [Online]. Available: https://doi.org/10.5555/927297

[62] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, pp. 676–682, 2012. [Online]. Available: https://doi.org/10.1038/nmeth.2019

[63] J.-C. Yen, F.-J. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370–378, 1995. [Online]. Available: https://doi.org/10.1109/83.366472

[64] W.-H. Tsai, "Moment-preserving thresolding: A new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377–393, 1985. [Online]. Available: https://doi.org/10.1016/0734-189X(85)90133-1

[65] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF Chimera—a visualization system for exploratory research and analysis," *Journal of computational chemistry*, vol. 25, no. 13, pp. 1605–1612, 2004. [Online]. Available: https://doi.org/10.1002/jcc.20084

[66] D. N. Mastronarde and S. R. Held, "Automated tilt series alignment and tomographic reconstruction in IMOD," *Journal of structural biology*, vol. 197, no. 2, pp. 102–113, 2017. [Online]. Available: https://doi.org/10.1016/j.jsb.2016.07.011

[67] C. Liu, L. Mendonça, Y. Yang, Y. Gao, C. Shen, J. Liu, T. Ni, B. Ju, C. Liu, X. Tang, J. Wei, X. Ma, Y. Zhu, W. Liu, S. Xu, Y. Liu, J. Yuan, J. Wu, Z. Liu, Z. Zhang, L. Liu, P. Wang, and P. Zhang, "The Architecture of Inactivated SARS-CoV-2 with Postfusion Spikes Revealed by Cryo-EM and Cryo-ET," *Structure*, vol. 28, no. 11, pp. 1218–1224.e4, 2020. [Online]. Available: https://doi.org/10.1016/j.str.2020.10.001

[68] L.-K. Huang and M.-J. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, no. 1, pp. 41–51, 1995. [Online]. Available: https://doi.org/10.1016/0031-3203(94)E0043-K

[69] C. Li and P. Tam, "An iterative algorithm for minimum cross entropy thresholding," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 771–776, 1998. [Online]. Available: https://doi.org/10.1016/S0167-8655(98)00057-9

[70] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985. [Online]. Available: https://doi.org/10.1016/0734-189X(85)90125-2

[71] C. Glasbey, "An analysis of histogram-based thresholding algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532–537, 1993. [Online]. Available: https://doi.org/10.1006/cgip.1993.1040

[72] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986. [Online]. Available: https://doi.org/10.1016/0031-3203(86)90030-0

[73] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. [Online]. Available: https://doi.org/10.1109/TSMC.1979.4310076

[74] W. Doyle, "Operations useful for similarity-invariant pattern recognition," *Journal of the ACM*, vol. 9, no. 2, p. 259–267, 1962. [Online]. Available: https://doi.org/10.1145/321119.321123

[75] A. Shanbhag, "Utilization of information measure as a means of image thresholding," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 5, pp. 414–419, 1994. [Online]. Available: https://doi.org/10.1006/cgip.1994.1037

[76] G. W. Zack, W. E. Rogers, and S. A. Latt, "Automatic measurement of sister chromatid exchange frequency." *Journal of Histochemistry & Cytochemistry*, vol. 25, no. 7, pp. 741–753, 1977. [Online]. Available: https://doi.org/10.1177/25.7.70454

[77] N. Vatani, F. Branch, R. Enayatifar *et al.*, "Gray level image edge detection using a hybrid model of cellular learning automata and stochastic cellular automata," *Open Access Library Journal*, vol. 2, no. 01, p. 1, 2015. [Online]. Available: https://doi.org/10.4236/oalib.1101203

[78] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013. [Online]. Available: https://doi.org/10.1007/978-3-662-05088-0

[79] C. K. Chow and T. Kaneko, "Automatic boundary detection of the left ventricle from cineangiograms," *Computers and biomedical research*, vol. 5, no. 4, pp. 388–410, 1972. [Online]. Available: https://doi.org/10.1016/0010-4809(72)90070-5

[80] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[81] N. Phansalkar, S. More, A. Sabale, and M. Joshi, "Adaptive local thresholding for detection of nuclei in diversity stained cytology images," in *2011 International Conference on Communications and Signal Processing*, 2011, pp. 218–220. [Online]. Available: https://doi.org/10.1109/ICCSP.2011.5739305

[82] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000. [Online]. Available: https://doi.org/10.1016/S0031-3203(99)00055-2

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3186146

Ngan Nguyen is a Ph.D. student at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. She received her M.Sc. degree from the University of Information Technology, Vietnam National University in Ho Chi Minh City, Vietnam. Her research focuses on computer graphics, visualization and their overlap with animation and biomechanics.

Ciril Bohak is a research scientist at King Abdullah University of Science and Technology (KAUST), Saudi Arabia, and an assistant professor in the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. He received his B.Sc., M.Sc., and Ph.D. degree from the University of Ljubljana. His research covers computer graphics, scientific visualization, and human-computer interaction.

Dominik Engel is a Ph.D. student at Ulm University, Germany, where he previously received his B.Sc. and M-Sc. degrees in computer science. In 2018 he joined the Visual Computing research group. His research focuses on deep learning in visualization and computer graphics, differentiable and neural rendering.

Peter Mindek is a post-doctoral researcher at TU Wien. He received his Ph.D. from TU Wien in 2015. His research interests include scientific visualization, storytelling, molecular graphics, and software architecture. He co-founded Nanographics, a startup developing technology for nanovisualization.

Ondřej Strnad is a research scientist at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. He received his Ph.D. from Masaryk University in Brno, Czech Republic, in 2014. His research interests span scientific visualization, geometry algorithms, and computer graphics. Recently, he joined the NANOVIS group at KAUST to work on technologies that deliver new visualizations and techniques for mesoscale biological models.

Peter Wonka is Full Professor in Computer Science at King Abdullah University of Science and Technology (KAUST) and Interim Director of the Visual Computing Center (VCC). Peter Wonka received his Ph.D. from the Technical University of Vienna in computer science. Additionally, he received a M.Sc. in Urban Planning from the same institution. After his Ph.D., he worked as a postdoctoral researcher at the Georgia Institute of Technology and as faculty at Arizona State University. His research publications tackle various topics in computer vision, computer graphics, remote sensing, image processing, visualization, and machine learning. The current research focus is on deep learning, generative models, and 3D shape analysis and reconstruction.

Sai Li received a B.Sc. degree in applied physics from Wuhan University, China in 2006, and Ph.D. in biophysics from the University of Goettingen, Germany, in 2012. He was a postdoctoral and senior research scientist at the Oxford Particle Imaging Centre, University of Oxford, U.K. from 2012 to 2018. He is currently an Associate Professor in the School of Life Sciences, Tsinghua University, China. His research interests include cryo-electron tomography, 3D reconstruction, and structural virology of pathogenic enveloped viruses.

Timo Ropinski is a professor at Ulm University, where he heads the Visual Computing Group. Before moving to Ulm, he was Professor in Interactive Visualization at Linköping University in Sweden, where he was the head of the Scientific Visualization Group. He received his Ph.D. in computer science in 2004 from the University of Münster, where he also completed his Habilitation in 2009. Currently, Timo serves as chair of the EG VCBM Steering Committee, and as an editorial board member of IEEE TVCG.

Ivan Viola is a Professor at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. He graduated from TU Wien, Austria, and in 2005 he took a post-doctoral position at the University of Bergen, Norway, where he was gradually promoted to the professor rank. In 2013 he received a WWTF grant to establish a research group at TU Wien. At KAUST, he continues developing new technologies that make visual, in-silico life at the nanoscale possible. Viola co-founded the Nanographics startup to commercialize nanovisualization technologies.