Tracing in 2D to Reduce the Annotation Effort for 3D Deep Delineation of Linear Structures

Mateusz Koziński[1], Agata Mosinska[2], Mathieu Salzmann, Pascal Fua

Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne, BC309, Station 15, CH-1015 Lausanne, Switzerland

Abstract

The difficulty of obtaining annotations to build training databases still slows down the adoption of recent deep learning approaches for biomedical image analysis. In this paper, we show that we can train a Deep Net to perform 3D volumetric delineation given *only* 2D annotations in Maximum Intensity Projections (MIP) of the training volumes. This significantly reduces the annotation time: We conducted a user study that suggests that annotating 2D projections is on average twice as fast as annotating the original 3D volumes.

Our technical contribution is a loss function that evaluates a 3D prediction against annotations of 2D projections. It is inspired by *space carving*, a classical approach to reconstructing complex 3D shapes from arbitrarily-positioned cameras. It can be used to train any deep network with volumetric output, without the need to change the network's architecture. Substituting the loss is all it takes to enable 2D annotations in an existing training setup. In extensive experiments on 3D light microscopy images of neurons and retinal blood vessels and on Magnetic Resonance Angiography (MRA) brain scans, we show that, when trained on projection annotations, deep delineation networks perform as well as when they are trained using costlier 3D annotations.

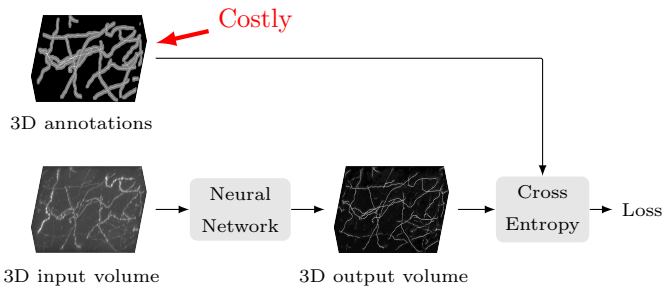
Keywords: Delineation, Segmentation, Deep Learning, Nerves, Vessels, Microscopy, Angiography

[1]M. Koziński was supported by the FastProof ERC Proof of Concept Grant
[2]A. Mosinska was supported by the Swiss National Science Foundation

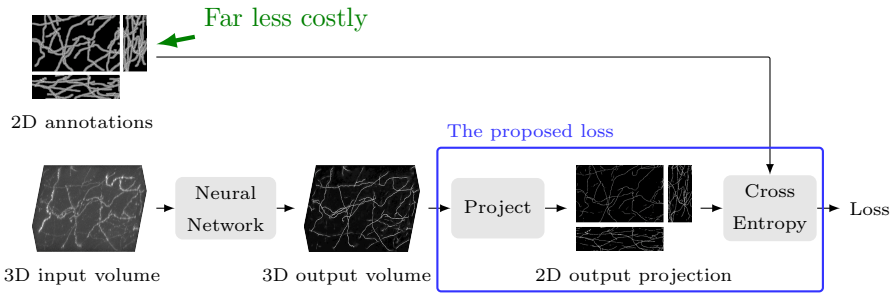
1. Introduction

Computed Tomography and Magnetic Resonance Scanners can now produce high resolution images, revealing fine details of vasculature for whole organs. This holds the promise of a systematic analysis of the vascular networks of the brain, or lung, which could open new diagnostic possibilities and give new insight into mechanisms underlying many diseases. For example, vasculature has been shown to carry information about brain tumor malignancy Bullitt et al. (2005), while vascular changes play an important role in pulmonary hypertension Rol et al. (2017), Shimoda and Laurie (2013), and in neurodegenerative diseases Sweeney et al. (2018). However, manually analyzing the finest structures of a whole organ is impractical due to the vast amounts of data that would need to be inspected. Similarly, modern microscopy techniques enable imaging neural networks at the scale of a whole mammalian brain. Unfortunately, due to the immense amount of data this produces, the feasibility of manual reconstruction is limited to tens of neurons while the ability to simulate signal propagation in artificially generated models of neural tissue reaches tens of thousands of neurons Markram et al. (2015). Automated reconstruction of connectivity maps would help bridge this gap, enabling simulation of large-scale models of real neural tissue.

Machine Leaning based techniques have demonstrated their effectiveness for the purpose of reconstruction of curvilinear networks, like vasculature and neural networks, but usually require substantial amounts of annotated training data to reach their full potential. Unfortunately, annotating complex topologies in 3D volumes by means of an inherently 2D computer interface is slow and tedious. The annotator must frequently rotate and move the volume to verify the correct placement of control points and to reveal occluded details. Not only is this inherently slow, but such interactions require continuously re-displaying large amounts of data, which often exceeds the capacity of a workstation, thus introducing further delays.



(a) Standard training procedure.



(b) Our training procedure.

Figure 1: Training a neural network to delineate 3D structures using 3D (a) and 2D (b) annotations. (a) The standard approach is to manually or semi-automatically delineate structures in 3D volumes to create ground-truth data, which can then be used to train a deep network. (b) Ours is to delineate in 2D in 2 or 3 Maximum Intensity Projections, which is easier and faster. The projections are used to compute a loss function that exploits these 2D annotations. We use it to train the network, and achieve similar performance with half as much human intervention.

In this paper, we show that we can train a Deep Net to perform 3D volumetric delineation given *only* 2D annotations in Maximum Intensity Projections (MIP), such as those shown on the left of Fig. 1. This is a major time-saver because delineating linear structures in 2D images is much easier than in 3D volumes and involves none of the difficulties mentioned above. Furthermore, semi-automated annotation tools work more smoothly on 2D than on 3D data. In short, limiting the annotation effort to the projections leads to a considerable labor saving without compromising the performance of the trained network.

More specifically, we introduce a loss function that penalizes discrepancies between the maximum intensity projection of the predictions and the 2D annotations. We show that it yields a network that performs as well as if it had been trained using full 3D annotations. The loss is inspired by *space carving*, a classical approach to reconstructing complex 3D shapes from arbitrarily-positioned cameras Kutulakos and Seitz (2000). Space carving exploits the fact that visual rays corresponding to background pixels in 2D images cannot cross any foreground voxel when passing through the volume. Conversely, rays emanating from foreground pixels have to cross at least one foreground voxel. In our case, the rays are parallel to the projection axes. The network is trained to minimize the cross-entropy between the 2D annotations and the maximum values along the rays.

Our contribution is therefore a principled approach to reducing the annotators' burden when training a Deep Net by enabling them to trace in 2D instead of 3D, while still capturing the full 3D topology of complex linear structures. We demonstrate this on 3D light microscopy images of neurons and retinal blood vessels and on Magnetic Resonance Angiography (MRA) brain scans. An earlier version of this approach first appeared in Koziński et al. (2018). We present here an extended version that includes a user study that demonstrates the effectiveness of our approach, as compared to more traditional ones.

2. Related Work

Delineation is a broad research topic. It operates on structures as different as roads (Mattyus et al., 2017, Mnih, 2013, Mnih and Hinton, 2010, Wegner et al., 2013), blood vessels (Ganin and Lempitsky, 2014, Maninis et al., 2016), bronchi (Meng et al., 2017), neurites (Peng et al., 2017, Sironi et al., 2016), and cell membranes (Mosinska et al., 2018), imaged using many different modalities. In this paper, we specifically address 3D delineation where the input is a volume, as opposed to a collection of ordered, but unregistered slices (Funke et al., 2012).

Early approaches to delineation of 3D curvilinear structures relied on filters

manually designed to respond strongly to tubular segments (Frangi et al., 1998, Law and Chung, 2008, Sato et al., 1998, Turetken et al., 2013). They do not require to be trained, but their performance degrades when the structures become irregular and the images noisy. This has led to the emergence of machine learning-based methods that can cope with such difficulties, given enough annotated data (Becker et al., 2013, Breitenreicher et al., 2013, Meng et al., 2017, Peng et al., 2017, Sironi et al., 2016). The most recent ones of these (Meng et al., 2017, Peng et al., 2017) rely on a combination of Deep Learning and adaptive exploration of the light microscopy images, and Computed Tomography (CT) scans.

However, using Machine Learning, and Deep Learning in particular, requires large amounts of annotated training data. Furthermore, annotating 3D stacks is much more labor-intensive than annotating 2D images. Only true experts, whose time is precious, are able to orient themselves and follow complex structures in large volumes (Peng et al., 2014). Until now, this problem has been handled by developing better ways to visualize and interact with image stacks (Peng et al., 2017, Vitanovski et al., 2009). Çiçek et al. (2016) annotated only a few slices of a volume and computed the loss using only them. The technique of Peng et al. (2014), like ours, allows the annotator to trace a linear structure in a maximum intensity projection and then attempts to guess the value of the third coordinate using a simple heuristic. While effective when the structures are relatively sparse, this can easily get confused as the scene becomes more cluttered.

There are numerous approaches to limiting the annotation effort associated to segmentation training include weak supervision in terms of scribbles (Can et al., 2018, Lin et al., 2016), bounding boxes (Dai et al., 2015, Khoreva et al., 2017, Rajchl et al., 2017, Shah et al., 2018, Zhao et al., 2018), image-level labels (Ahn and Kwak, 2018, Jing et al., 2018, Papandreou et al., 2015, Pinheiro and Collobert, 2015), or any combination thereof. They often involve iterative estimation of the unknown full annotations together with network parameters in an Expectation-Maximization-like procedure, where additional prior knowledge

is specified in form of a Markov Random Field. It has been shown that, for some tasks, networks trained with weak supervision attain performance very close to that of fully supervised networks (Can et al., 2018). These approaches to making 2D annotation easier and faster could be used in conjunction with our approach, resulting in a further decrease of the labeling workload.

In most existing 3D segmentation algorithms we know of, training is performed using the standard cross entropy. Specific connectivity loss functions have been designed for reconstructing neural morphologies from electron microscopy data (Briggman et al., 2009, Funke et al., 2018) with the goal of promoting correct connectivity of the cells over spatial segmentation accuracy. By contrast, our loss function is intended for enabling training on 2D annotations.

The originality of our approach is to introduce a method that relies solely on 2D annotations in Maximum Intensity Projections, yet captures the 3D structure of complex linear structures when the projections are used jointly.

3. Method

3.1. From 3D to 2D Annotations

Let us first consider the problem of training a neural network f_w, parameterized by weights w, to segment linear structures within 3D image stacks, given a training set T of pairs $(\mathbf{x}, \tilde{\mathbf{y}})$, where each 3D image $\mathbf{x}$ is accompanied by the corresponding volumetric ground-truth annotations $\tilde{\mathbf{y}}$. We denote the elements of $\mathbf{x}$ and $\tilde{\mathbf{y}}$ by x_{ijk} and $\tilde{y}_{ijk}$, where i, j, k index the positions of the elements within the volumes. The ground-truth labels take a value in the set $\{1, 0, \varnothing\}$, which indicate the presence of a linear structure in voxel i, j, k if $\tilde{y}_{ijk} = 1$, the absence of a linear structure if $\tilde{y}_{ijk} = 0$, and uncertainty of the annotator if $\tilde{y}_{ijk} = \varnothing$. Delineation can then be cast as a binary segmentation problem by simply ignoring the voxels labeled as $\varnothing$ during training. The network output $\mathbf{y} = f_w(\mathbf{x})$ has the same size as the input and contains probabilities of presence

of a linear structure in each voxel. To train the network, we find

$$\arg\min_{w} \sum_{(\mathbf{x},\tilde{\mathbf{y}})\in T} \sum_{i,j,k} L(f_w(\mathbf{x})_{ijk}, \tilde{y}_{ijk})\,, \tag{1}$$

where $f_w()_{ijk}$ denotes voxel i, j, k of the prediction, and the loss $L(y, \tilde{y})$ is taken to be the cross entropy $C(y, \tilde{y}) = [\tilde{y} = 1] \log y + [\tilde{y} = 0] \log(1 - y)$, where $[\cdot]$ is the Iverson bracket. As discussed in the introduction, the drawback of this approach is that generating the ground-truth labels $\tilde{\mathbf{y}}$ in sufficient numbers to train a deep network is tedious and expensive when operating on large volumes.

To alleviate this problem, we reformulate the loss function of Eq. 1 so that it can exploit annotated Maximum Intensity Projections (MIPs) of the input volumes. A MIP of volume $\mathbf{x}$ along direction i, which we denote as $\mathbf{x}^{\mathrm{i}}$, is a 2D image with elements $x^{\mathrm{i}}_{jk} = \max_i x_{ijk}$. Annotating MIPs is easy when the structures of interest have high intensity and are clearly visible in the projections. A MIP annotation $\tilde{\mathbf{y}}^{\mathrm{i}}$ of the projection $\mathbf{x}^{\mathrm{i}}$ is a 2D image with elements $\tilde{y}^{\mathrm{i}}_{jk} \in \{1, 0, \varnothing\}$, where the labels have the same interpretation as the ones used for annotating in 3D. MIPs of the volume along the directions j and k, and their annotations, are defined similarly.

The key property of MIP annotations, is that $\tilde{y}^{\mathrm{i}}_{jk} = 0$ tells us that *all* voxels of the input column jk contain background. To see that the property really holds, let us assume an idealized case where the Maximum Intensity Projection operation, and the act of annotation, preserve the linear structures. In other words, we assume that, if the training volume contains an image of a linear structure in any voxel of column jk, then this linear structure will necessarily be visible in the Maximum Intensity Projection, in pixel $\mathbf{x}^{\mathrm{i}}_{jk}$, and will be annotated as foreground in the MIP annotation, so that $\tilde{y}^{\mathrm{i}}_{jk} = 1$. Under these assumptions, by De Morgan's law, $\tilde{y}^{\mathrm{i}}_{jk} \neq 1$ implies that no voxel of the column jk is of foreground class.

It is exactly this property that enables establishing a link between training on MIP annotations and space carving. In space carving, a single background pixel of an image of a 3D scene is used to classify many voxels of scene reconstruction as background, effectively carving out the reconstructed shape. When training a

network on MIP annotations, a pixel annotated as background could be used to constrain many voxels of the prediction to belong to the background class, thus generating an error signal for these voxels. In practice, instead of enforcing this constraint directly, we formulate a loss function that capitalizes on this observation implicitly. To that end, we define the max-projection $f_w^{\mathrm{i}}(\mathbf{x})$ along direction i of the network output as the image with elements $f_w^{\mathrm{i}}(\mathbf{x})_{jk} = \max_i f_w(\mathbf{x})_{ijk}$. We proceed similarly for directions j and k. We then define the loss as

$$\sum_{(\mathbf{x},\tilde{\mathbf{y}})\in T} \Big(\sum_{jk} L\big(f_w^{\mathrm{i}}(\mathbf{x})_{jk}, \tilde{y}_{jk}^{\mathrm{i}}\big) + \sum_{ik} L\big(f_w^{\mathrm{j}}(\mathbf{x})_{ik}, \tilde{y}_{ik}^{\mathrm{j}}\big) + \sum_{ij} L\big(f_w^{\mathrm{k}}(\mathbf{x})_{ij}, \tilde{y}_{ij}^{\mathrm{k}}\big) \Big). \quad (2)$$

To see the analogy to space carving, note that, by its definition, $f_w^{\mathrm{i}}(\mathbf{x})_{jk}$ upper bounds the predicted probability of presence of a linear structure in column jk. Eq. 2 penalizes large values of this upper bound whenever $\tilde{y}_{jk}^{\mathrm{i}} = 0$. In other words, a single background label in a 2D annotation results in minimization of a whole column of predictions, mimicking space carving. When $\tilde{y}_{jk}^{\mathrm{i}} = 1$, minimizing the loss increases the largest prediction in the column. The latter one might be placed off a linear structure, but it is then likely to be penalized by a component of the loss defined for another projection.

As only the maximal element in each row, column, and tube contributes to the predicted projection, the derivatives of the loss (2) with respect to the predictions are zero for all the elements of the volume, except for the maximal ones. In other words, the gradient tensor of the loss is very sparse. In theory, this should detract from the effectiveness of the gradient-based training procedure. In practice, the nonzero elements are not distributed randomly over the gradient tensor, but penalize the strongest wrong predictions in their rows, columns, and tubes, as explained above. As will be shown in sections 4.2.3 and 4.3, the networks trained with 2D annotations perform on par with ones trained on the full 3D annotations and the space carving mechanism seems to be the secret behind this surprising result. This hypothesis is supported by the fact that, in our experiments, networks trained on slice annotations deliver inferior performance even though the loss gradient is equally sparse in both methods.

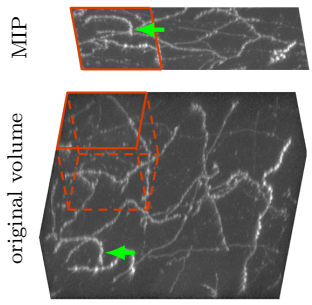


Figure 2: When training on MIP annotations, using volume crops (brown cube) may lead to situations where, a crop of a MIP annotation (brown rectangle) contains labels of linear structures from outside of the volume crop (marked with green arrows). This annotation noise could adversely influence performance of the trained network.

3.2. Visual Hull for Training on Cropped Volumes

Due to memory limitations, the annotated training volumes are typically cropped into sub-volumes and the MIP annotations can be cropped to match. However, the cropped annotations may then contain labels for structures located outside the volume crop, as illustrated by Fig. 2. To reduce the influence of these extraneous annotations, we use another element of the space carving theory, the visual hull $\mathbf{h}$. $\mathbf{h}$ is a volume containing the original one, and constructed from its projections (Kutulakos and Seitz, 2000). A toy example of a visual hull created from 2D projections of a volume is presented in Fig. 3(a). We define it more precisely below.

We first introduce the definition of the hull for the classic, binary case. Given three orthogonal MIP annotations $\tilde{\mathbf{y}}^{\mathrm{i}}, \tilde{\mathbf{y}}^{\mathrm{j}}, \tilde{\mathbf{y}}^{\mathrm{k}}$, with elements $\tilde{y}^{\mathrm{i}}_{jk}, \tilde{y}^{\mathrm{j}}_{kk}, \tilde{y}^{\mathrm{k}}_{ij} \in \{0, 1\}$, we define the hull $\mathbf{h}$ as a binary volume with elements

$$h_{ijk} = \begin{cases} 1 & \text{if } \tilde{y}^{\mathrm{i}}_{jk} = 1 \wedge \tilde{y}^{\mathrm{j}}_{ik} = 1 \wedge \tilde{y}^{\mathrm{k}}_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

By construction, an element of the hull $h_{ijk} = 1$ if and only if *all* of its projections are labeled as foreground. In our context, a foreground voxel outside a crop only produces an incorrect label in *a single* projection, as demonstrated in Fig. 2. As shown in Fig. 3(b), we can eliminate such false positive labels by

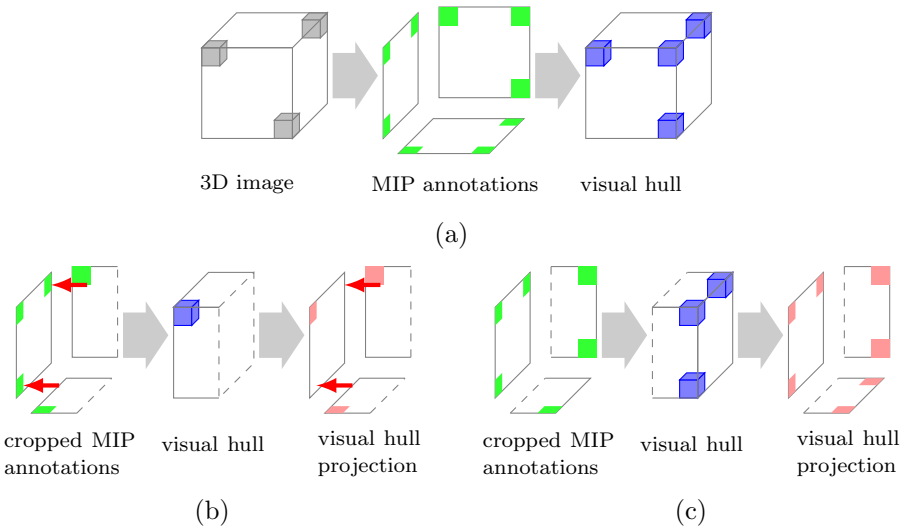


Figure 3: Handling cropped volumes. *(a)* A 3D volume with three foreground voxels, the annotations of its MIPs in green, and the visual hull computed from these in blue. *(b)* The volume has been cropped so that only the left half remains. The annotations have been cropped to match, leaving a single blue voxel in the visual hull. Reprojecting it into the MIPs lets us eliminate the extraneous annotations, indicated with red arrows. *(c)* However, there are situations such as the one depicted here, where some will survive.

projecting the visual hull back to the 2D annotations and discarding the labels that fall outside of the projection of the visual hull. However, this technique fails to eliminate these false positive labels, for which in each of the remaining projection annotations another positive label exists with the same coordinate along the common dimension. Such situation is illustrated in Fig. 3(c). Our experiments show that such rare events have little impact on the performance of the trained network.

As stated in section 3.1, in practice our annotations are defined in terms of a ternary set of labels, with the additional label $\varnothing$, allowing the annotator to skip labeling a pixel if he is not certain of its class. In our experiments, we also use this additional label to create margins around thin annotations of centerlines of linear structures, in order to account for the ambiguity in defining the latter. In order to apply the visual-hull-based technique to eliminate false positive

labels from such ternary MIP annotations, we reduce the number of classes in the annotations to two when constructing the visual hull. More precisely, we consider the foreground label and the label encoding the uncertainty of the annotator as positive, and the background label as negative. Then, for each projection annotation, we project the hull along the same direction and suppress all the positive and uncertainty labels that collide with the negative class in the projection of the hull. In other words, we propagate the background labels between projections via the visual hull.

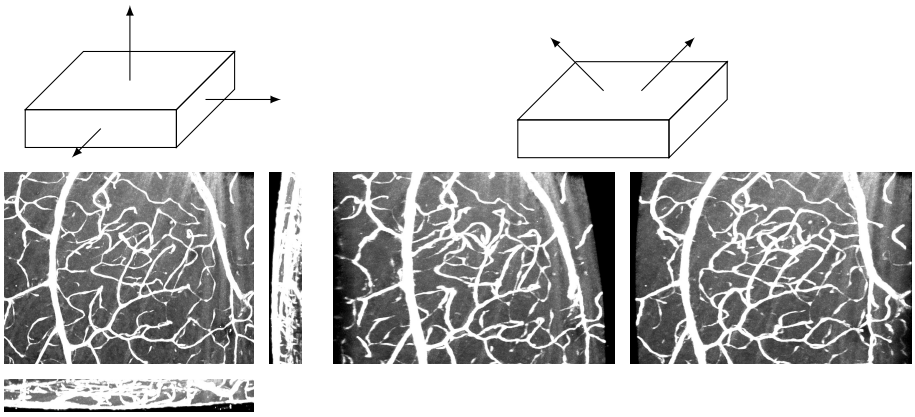
In the experiments presented in section 4 we train a deep network on 3, 2 or 1 MIP annotations per volume. The definition of the visual hull presented above trivially generalizes to the 2-MIP cases, and the procedure is not performed when only 1 MIP annotation is used.

3.3. Limitations of our Method.

As mentioned above, the main requirement for our approach to be effective is that the target structures be clearly visible in the projections, so that 2D annotation is faster and easier than its 3D counterpart. This property is hard to quantify but easily assessed by visual inspection of the data. Ideally, the structures of interest should be brighter or darker than other structures visible in the image, which is the point of most staining techniques. However, the presence of a few bright background objects occluding small portions of the structures of interest is typically not an issue because our approach is robust to small disruptions in the annotations.

In addition to being visible, the target structures must have 3D shapes such that their 2D projection is informative. This excludes structures with extensive self-occlusions or numerous holes and concavities. In difficult cases, it would help to split the volume into smaller chunks before projecting and to find projection directions that best reveal the structures of interest, as will be discussed in the following section.

In Section 4, we will show that our training method is effective for the delineation neurons and blood vessels in 2-photon and confocal microscopy as well as



axial projections oblique projections

Figure 4: *Left:* axial projections of a confocal microscopy image of retinal veins. Two of the three projections are very cluttered. *Right:* a pair of orthogonal, non-axis-aligned projections of the same volume, much easier to annotate. The actual projection directions are visualized above.

in Magnetic Resonance Angiography volumes. Additional examples of suitable data include Tomography images of airways and lung vasculature in thoracic CT images. By contrast, we found annotating mitochondria in projections of electron microscopy images very difficult.

3.4. Projections along Arbitrary Directions

In some modalities, the volumic distribution of target structures is anisotropic, either due to scale differences across directions or due to specificities of the imaged specimen itself. For example, the retinal veins shown in Fig. 4 are densely packed along the z direction. This results in many occlusions with one vein hiding the other and makes the axial projection much harder to annotate. As stated above, one way around this problem is to split the volume into smaller subvolumes and annotate the less cluttered projections of the subvolumes. However, that increases the number of projections that require annotation and therefore the time-saving our method can deliver. Another way is to project the volume along directions that do not align with coordinate axes. We will show in the results section that this yields a substantial performance improvement.

When done naively, projecting a volume along an oblique direction involves rotating it in 3D and projecting its rotated version. In the case of sandwich-shaped volumes whose z dimension is much smaller than the other two, like the retina of Fig. 4, storing the rotated copies require much more memory than storing flattish originals. Rather than rotating and projecting, we therefore perform the projection by tracing lines through the original volume. We trace one line through each voxel of these faces of the volume, the normals of which give a positive scalar product with the projection direction. For each such line, we retain the largest of all the voxel values that the line traverses as the color of the projection. Like ordinary max-projection, this projection mechanism is differentiable, and can be used to compute our loss function (2).

Performing the tracing line-by-line requires random access to the input volume, which could slow down the computation of the projections. We therefore perform the projection voxel-by-voxel. More precisely, we first initialize an empty projection, and then, for each voxel of the volume, compute the pixel to which it projects. We set the pixel to the minimum of its current value and the value of the projected voxel. This algorithm is easily parallelizable and can be used for computing projections of volumes that are too large to fit in the memory, by processing a single slice at a time.

We selected the projection direction manually. For the data presented in Fig. 4, where the vessels lie roughly in the xy plane, a pair of orthogonal directions inclined at $\angle 45$ to the z-axis exposed the vessels well. However, a method for automatic identification of the best projections to annotate would help handling the use cases where very large volumes are stored on a remote server, making interactive selection prohibitively slow, or where the imaged structures have very complex topology, making the choice of the optimal projections confusing. We leave the exploration of this path for future research.

3.5. Implementation

In practice, we implemented f_w as a U-Net style network (Ronneberger et al., 2015). Specifically, we used the network presented in Fig. 5. We only

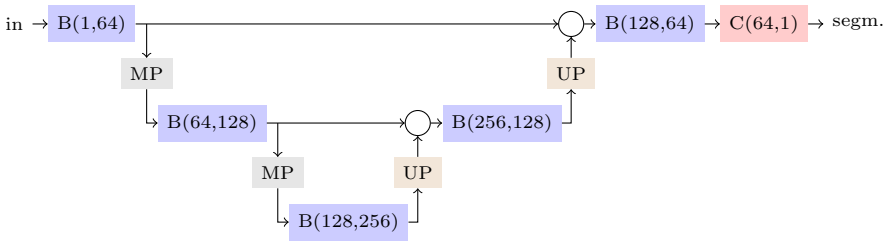


Figure 5: The U-Net-style architecture used in our experiments. $\mathrm{B}(n_{\mathrm{in}}, n_{\mathrm{out}})$ denotes a block that includes a convolution with n_{in} input features and n_{out} output features, Batch Normalization, ReLU, another convolution with n_{out} features both in the input and output feature map, Batch Normalization, ReLU and spatial dropout with probability 0.1. MP denotes max pooling in windows of diameter 2, and stride 2. UP denotes a sequence of a convolution with two times less output than input features and stride 2. A circle denotes concatenation of its inputs. $\mathrm{C}(n_{\mathrm{in}}, n_{\mathrm{out}})$ denotes a convolution with n_{in} input and n_{out} output features. The receptive field of this network has a diameter of 44 voxels.

used two max-pooling operations instead of the usual four, which resulted in a more compact network that fits in memory even with larger volume crops. In all our experiments, we trained the network for 200K iterations, using the ADAM update scheme (Kingma and Ba, 2015) with momentum of 0.9, weight decay 10^{-4} and step size 10^{-5}.

4. Experimental Evaluation

4.1. Datasets

We tested our approach on four data sets that differ in terms of the imaged tissue, the acquisition modality and the image resolution. There are substantial variations between these datasets with respect to the density of the structures of interest, their appearance, and the amount of clutter originating from extraneous objects. Together, they constitute an exhaustive benchmark for 3D delineation.

Axons. The dataset comprises 16 stacks of 2-photon microscopy images of mouse neural tissue, with sizes ranging from $40 \times 200 \times 200$ to $136 \times 322 \times 500$ voxels and a resolution of $0.8 \times 0.26 \times 0.26$ µm. The images were acquired in

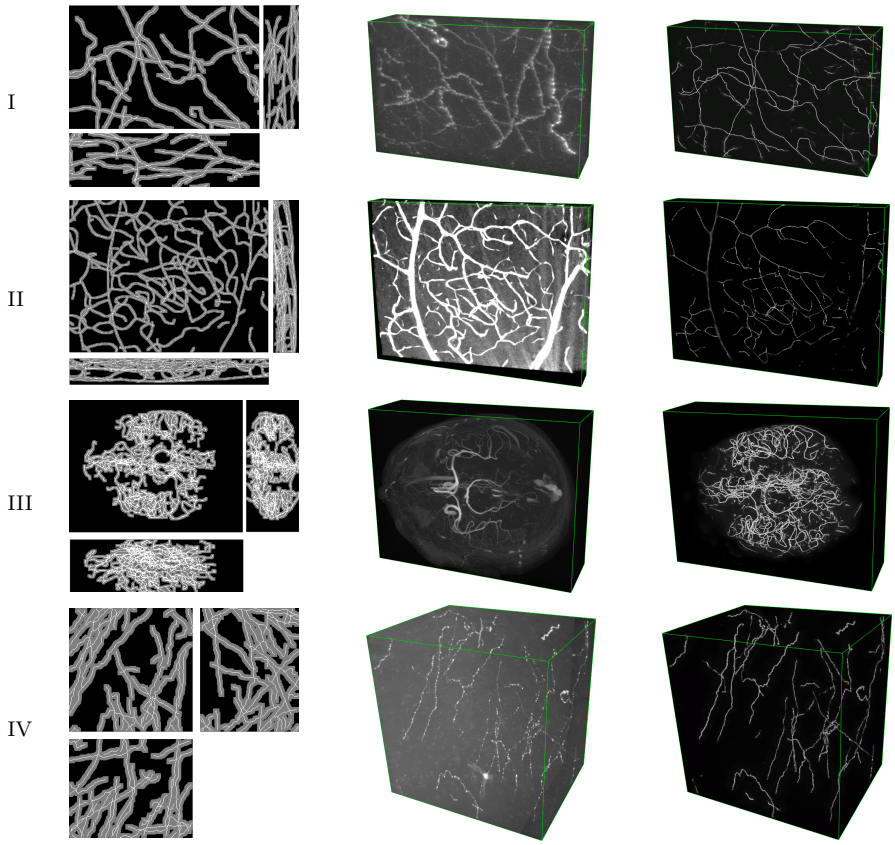


Figure 6: Results on our four datasets, from top to bottom, axons, retinal blood vessels, brain vasculature in MRA scans, and neural tissue in mouse brain. (a) 2D annotations in 3 MIPs of a test volume. The foreground centerline annotations are marked in white and the regions to be ignored around them in gray. (b) Input test image volume. (c) Output segmentation.

vivo, from a mouse with a translucent window implanted in the scalp. We split the data into a test set of two volumes of size $136 \times 233 \times 500$, and a training set of 14 smaller volumes. The top row of Fig. 6 depicts one of the test volumes.

Retina. The dataset is made of two confocal microscopy image stacks depicting retinal blood vessels. The stacks have a size of $1024 \times 1024 \times 110$ voxels and a resolution of 0.62 µm. We use one of them for training and the other, depicted in Fig. 6, for testing. Since most vessels are located within a 50-pixel high XY

slice, MIPs in the X and Y directions are very cluttered. Therefore, we split the volume into 16 subvolumes, sized $256 \times 256 \times 110$ voxels, and annotated their MIPs. In other words, we also traced the vertical faces of the smaller volumes. This only requires annotating 6 additional 1024×110 images, which is still fast. The middle row of Fig. 6 describes both our 2D annotations and the segmentation results for the test volume.

Angiography. This set of MRI brain scans (Bullitt et al., 2005), one of which is shown in Fig. 6, is publicly available. It consists of 42 annotated stacks, which we cropped to a size of $416 \times 320 \times 128$ voxels by removing their empty margins. Their resolution is $0.5 \times 0.5 \times 0.6$ mm. We randomly partitioned the data into 31 training and 11 test volumes. As in the case of the retinal vessels, we decreased the visual clutter by splitting each volume into four $208 \times 160 \times 128$ subvolumes for which we produced 2D annotations. This requires annotating an additional 416×128 image and a 320×128 one for each training volume. The bottom row of Fig. 6 describes both our 2D annotations and our results on one of the test stacks.

Brain. The dataset is a part of a 2-photon microscopy scan of a whole mouse brain. It contains 14 stacks of size $250 \times 250 \times 200$ voxels and a spatial resolution of $1.0 \times 0.3 \times 0.3$ µm. Compared to the Axons dataset the volumes are more diverse since they were pooled randomly from different brain regions. We use 10 stacks for training and 4 for testing. The last row of Fig. 6 depicts an example volume.

All the manual annotations are expressed in terms of 2D and 3D centerlines of the underlying structures. We then use a pixel-width of 11 for Axons, Retina and Brain datasets, and 7 for the Angiography volumes, to define the area to ignore around the centerline when computing the loss, as discussed in Section 3.1, as well as to compute the visual hulls, as described in Section 3.2.

Table 1: The total time needed to complete annotations of the whole Mouse Brain dataset in the user study.

Annotation method	Annotation time		Performance
	[min]	[% 3D time]	[Dice score]
Annotating in 3D	609	100	80.2
Annotating 3 2D MIPs	387	64	80.0
Annotating 2 2D MIPs	277	45	80.0
Annotating 1 2D MIP	152	25	49.2

4.2. User Study

The usefulness of our approach is predicated on the claim that annotating linear structures in 2D is much easier than doing it in 3D, while the two annotation types give equally good results when used for training. To substantiate this claim, we conducted a user study involving 15 PhD students used to performing such delineation for research purposes. We asked them to annotate one volume from the Brain dataset in 2D, and another one in 3D. The annotation was performed using the Fiji Simple Neurite Tracer plugin (Frangi et al., 1998). We present the analysis of the data collected in the study below. In subsection 4.2.1 we demonstrate that switching to annotating in 2D enables annotating the data set twice as fast as in 3D. In subsection 4.2.2 we show that, the 2D annotations are nevertheless consistent with the 3D ones. Finally, in subsection 4.2.3 we demonstrate that, when used for training with our method, they yield networks performing on par with ones trained on the full 3D annotations.

4.2.1. Efficiency of MIP annotation

To estimate the annotation workload we recorded the wall-clock time it took the participants to complete their tasks. We present the results in Fig. 7. Annotating three projections per volume was quicker than performing full three-dimensional annotations for all but two volumes, and annotating just two projections was at least two times quicker for all but four volumes. The few cases where annotating projections took longer show that individual times are not a reliable measure of annotation efficiency, as they include a high dose of random-

ness. They are influenced by many factors, including personal predispositions, familiarity with the task and the tool, whether the annotator was asked to perform the 3D or the 2D annotation first, and random events, for example, loosing concentration or a crash of the annotation tool. However, meaningful patterns emerge from the data as a whole. This is best illustrated in Table 1, containing aggregated times. Annotating all 15 volumes in 3D took the participants of our user study 10 working hours in total. The time needed to label the dataset in 2D was 6.5 hours, or 65% of the 3D annotation time, when annotating 3 MIPs per volume. This could further be reduced to 45% by annotating only two projections per volume, and to 25% by annotating only one. The differences in the average time needed to annotate each of the views stem from the non-isotropy of the data. The scans have lower resolution along the z axis than in the xy-plane. Additionally, the sizes of the annotated volumes along these dimensions differ.

4.2.2. Quality of the 2D annotations

The results of our user study suggest that annotating a dataset in 2D requires two times less work than doing it in 3D. But are the 2D annotations equally good as the 3D ones: 2D projections carry less information than the original 3D data and one might wonder if this affects the quality of the 2D annotations. To answer this, we evaluated the quality of the 2D projection annotations produced in our user study by comparing them to the 3D annotations. More precisely, we projected the 3D annotations and compared the 2D MIP annotations to the resulting projections. We computed the precision P and recall R of the 2D annotations with respect to the projections of the 3D annotations, defined as $P_{\text{2D3D}} = \frac{\sum_{ij}[\tilde{\mathbf{y}}_{ij}^{\text{2D}}=1][\tilde{\mathbf{y}}_{ij}^{\text{3D}}=1]}{\sum_{ij}[\tilde{\mathbf{y}}_{ij}^{\text{2D}}=1]}$ and $R_{\text{2D3D}} = \frac{\sum_{ij}[\tilde{\mathbf{y}}_{ij}^{\text{2D}}=1][\tilde{\mathbf{y}}_{ij}^{\text{3D}}=1]}{\sum_{ij}[\tilde{\mathbf{y}}_{ij}^{\text{3D}}=1]}$, where $[\cdot]$ is the Iverson bracket, and the summation is over all pixels of the projection. We found $P = 75\%$ and $R = 70\%$ indicating reasonable consistency. Given that the annotations are one-pixel-thick centerlines, some of the inconsistent annotations might simply be shifted by a small distance, while others may be missing altogether. To investigate this, we checked what percentage of annotations of one type is within a distance of no more than d pixels to the closest annotation

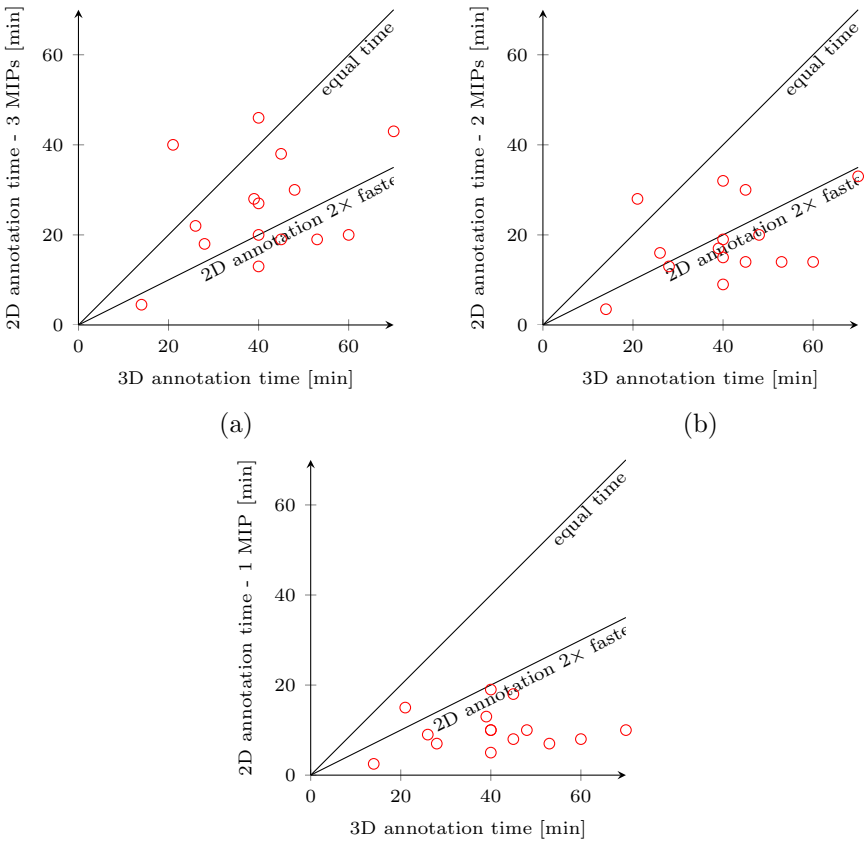


Figure 7: Annotation times captured during the user study. The volumes of the mouse brain dataset were annotated both in 3D and in 2D by different users to ensure that the users are not familiar with the stack they were annotating. A pair of annotation times is represented as a single point in each of the plots. Plot (a) presents the time needed to annotate 3 MIPs in 2D, the time needed to annotate 2 MIPs is presented in plot (b), and plot (c) depicts the amount of time necessary to annotate 1 MIP for each training volume.

of the other type. The results are presented in Fig. 8. We vary d between 1 and 10 and observe that over 95% of all 2D annotations are within a distance of 3 pixels from a projection of a 3D annotation, and vice versa. The results suggest that less than 5% of annotations of each type are inconsistent with the annotations of the other type.

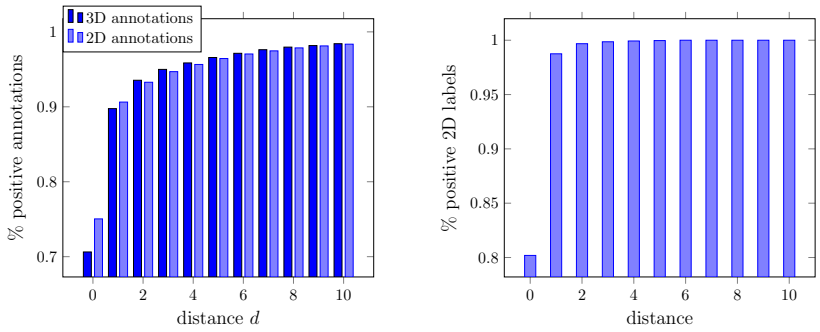
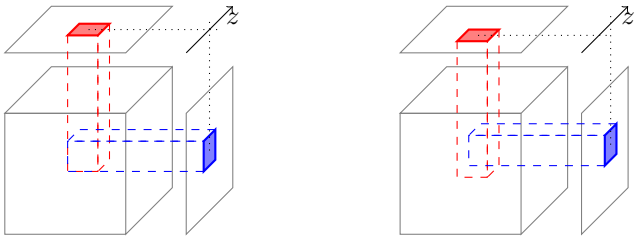


Figure 8: *Left*: Consistency of the 2D and 3D annotations produced for the Brain dataset in our user study. The bars show the percentage of positive 2D labels within a distance d to the closest projection of a positive 3D label, and the percentage of projections of 3D labels within a distance of d to the closest 2D label, as a function of d. 95% of positive annotations of each type have a corresponding positive annotation of the other type within a distance of less than three pixels, indicating the generally high consistency between the 2D annotations and the 3D ones. *Right*: An estimate of the percentage of 2D projection annotations inconsistent across different projections of the same volume. The bars represent the fraction of 2D annotations that have a corresponding annotation in another view at a distance of at most d pixels, as a function of the distance d. 20% of the annotations appear to be inconsistent, but almost never by more than three pixels.

We have shown that the 2D projection annotations are roughly consistent with the 3D annotations. However, since the 2D annotations are performed independently for different projections, inconsistencies may still occur between the 2D annotations of different projections of the same volume. More precisely, each pair of projections of a 3D volume has one dimension in common. Annotations of the two projections are consistent, if for a foreground voxel of the volume, the corresponding foreground pixels in both projection annotations have the same coordinate along the common dimension. The concept is illustrated in Fig. 9. To quantify the inconsistency of the annotations resulting from our user study, we build up on the fact that a pair of isolated, inconsistent 2D annotations, like the ones presented in Fig. 9, creates an empty visual hull. Therefore, the number of inconsistent 2D annotations can be estimated by constructing a visual hull from the 2D projection annotations, projecting the hull back to 2D and



(a) (b)

Figure 9: A pair of consistent (a) and inconsistent (b) MIP annotations. A single pixel has been annotated as foreground in each of the two projections (in red and blue). Consistent annotations co-occur along the common dimension (z), while the inconsistent annotations do not. For inconsistent annotations, the gradient of our loss function is distributed over a larger number of voxels. The analysis of consistency of annotations performed independently for different projections is presented in section 4.2.

counting the number of positive labels that fall outside of the hull projection. As in the case of the hull-based filtration introduced in section 3.2, projections of the hull may fail to eliminate some inconsistent annotations, which means that the resulting estimate is a lower bound. Additionally, we can estimate the degree of inconsistency by verifying how much the position of the inconsistent annotations differs along the common dimension. That is, we estimate how many annotations are inconsistent by no more than a given distance d by dilating the annotations with a structuring element of radius d before constructing the hull. The results are presented in Fig. 8. This procedure confirms that at least 20% of the annotations are inconsistent, but almost never by more than 3 pixels. The effect of training on inconsistent annotations is that the error signal that is focused on a single voxel when the annotations are consistent, gets distributed over a larger number of voxels. However, as demonstrated below, the performance attained by training on MIP annotations in the experiments appears not to be affected by this level of inconsistency.

4.2.3. Performance

We have asserted the high quality of 2D annotations, and we now confirm their utility for training a Deep Net. As stated above, even though they are

highly consistent with the 3D annotations, they *do* contain less information. Moreover, as explained in section 3.1, the proposed 2D loss function (2) yields very sparse gradients with respect to the 3D output of the network, with only a single nonzero value in each row, column, or tube. It is not clear *a priori* that such sparse error signals are equivalent to the dense gradients obtained on full 3D annotations. To verify this, we compared performance of networks trained on the two types of annotations. We express the performance in terms of the maximum Dice score—the harmonic mean between the precision and recall—a standard metric for binary segmentation evaluation, also called the F1 score. In Table 1, we report these scores when using either 3D annotations or 2D annotations, in three, two, or only one MIP. For training with a single MIP, we used the annotation of the z-projection, (the top-left images in the leftmost column of Fig. 6), and added the y-projection (bottom images in the same figure) when training on two MIPs. When using three, or even only two MIPs, there is virtually no performance loss for a reduction in annotation time of 36, and 55%, respectively. This validates our claim that the level of inconsistency in the annotations, exposed in section 4.2.2 does not affect the final performance: a network trained on the partly inconsistent projection annotations almost matches the network trained on full 3D annotations in terms of performance. This surprisingly high performance in spite of the sparsity of gradients our method yield can be explained by analogy to space carving as mentioned in section 3.1. The method finds its limits when we annotate only one MIP, which results in a severe performance drop. This makes intuitive sense because, for reasonably simple shapes, space carving can yield informative estimates from only two views but not from a single one.

4.3. Further Quantitative Evaluation

In the user study of Section 4.2 the 2D and 3D annotations were generated independently. We demonstrated that they were roughly equivalent, yielding networks of similar performance when used for training. To evaluate the proposed approach more extensively, for various imaging modalities and specimens,

Table 2: Performance and the corresponding time savings.

	Dice score				Time saved[a] [%]
	Axons	Retina	Angiography	Mouse	
UNet/3D annot.	75.4	**81.5**	**77.6**	80.2	0
UNet/3 MIP per volume	**78.1**	78.2	75.9	**82.2**	35
UNet/2 MIP per volume	75.0	77.8	74.8	80.0	55
arbitrary projections[c]	—	80.8	—	—	—
UNet/1 MIP per volume	72.3	39.0	57.7	50.1	70
Turetken et al. (2013)	58.8	77.1	22.7	18.1	100
Çiçek et al. (2016)	70.8	75.8	74.1	67.5	35[b]
Sironi et al. (2016)	68.5	62.6	50.3	53.6	0

[a] The perc. of time saved w.r.t. 3D annotation, as estimated in the user study.

[b] Slice annotation was assumed to be equally time-consuming as MIP annotation.

[c] Training on annotations of the two non-axial projections from Fig. 4.

we perform experiments on the three remaining data sets. Instead of performing the 2D annotations from scratch, we now use projections of the 3D annotations as the annotations of 2D projections. This is not what we would do in practice but it guarantees that their quality is *exactly* the same, while still enabling to test the basic concept of training on less informative 2D annotations, with a loss function yielding extremely sparse gradients.

We report our results in Table 2. In the rightmost column, we give an estimate of the time saved by generating the 2D annotations instead of the 3D ones on the basis of the above user study. When training on 3 or 2 MIPs per volume, we obtain roughly the same results as when training on full 3D annotations—slightly better for the Axons and Brain, and slightly worse for the Retina and Angiography datasets—while, as shown in section 4.2, the corresponding annotation effort is decreased by 45 and 55 percent, respectively. Note that in the Retina case, training on the annotations of the two less cluttered, non-axial projections of the Retina yields better results than training on the cluttered axial projections. This demonstrates the utility of annotating well-chosen projections of non-isotropic data.

In short, training on 2 MIP annotations per volume enables attaining the

same precision as training on the full 3D annotations, but at half of the annotation cost. These results are fully consistent with the findings of the user study presented above. While offering further time saving, the reduction of the amount of annotations used to a single projection per volume leads to a substantial performance drawback. We leave it for future work to investigate possible methods of preventing this adverse effect.

Whether using 3D or 2D annotations, these results rely on the modified U-Net architecture discussed in Section 3.5. For completeness, we also list in Table 2 the performance of three earlier methods. One alternative method of limiting annotation effort required to train a volumetric Deep Net is to annotate a small subset of slices of the original volume (Çiçek et al., 2016). In our experiments, the number of annotated slices used to train the network using this approach exactly matched the number of projections used in our method. We always annotated axis-aligned slices in the middle of the volumes. For a fair comparison, we also used the same network architecture in the two sets of experiments. While for the Retina and Angiography datasets the performance of a network trained on slice annotations closely matched that of the network trained on MIP annotations, the performance gap is larger for the two datasets depicting neural tissue. Moreover, it is often the case that the topology of linear structures is more easily disambiguated in the projections than in isolated slices, which makes annotating the projections easier. We also compare the performance of our method to a hand-crafted tubular structures detector (Turetken et al., 2013) that does not require any annotations. Not surprisingly, it performs well on the Retina dataset, used by the authors to develop the technique, but fails to generalize to the other datasets, not considered when designing the detector. The last baseline used in the experiments is a regression-based approach to delineation (Sironi et al., 2016), trained on the original set of 3D annotations, which our approach also outperforms. Its inferior performance might stem from the fact that the GradientBoost algorithm at the heart of the approach is less powerful than our neural network.

5. Conclusion

We have proposed a method for training DNNs to segment 3D images of linear structures using only annotations of 2D maximum intensity projections of the training data instead of full 3D annotations. We demonstrated that this results in decreased annotation requirements without loss of performance. To this end, we have exploited properties of visual hulls that are not specific to linear structures. In future work, we therefore intend to show that the scope of this technique is in fact much broader, for example by applying it to 3D membrane extraction. We also plan to extend our approach by developing an automated method for selecting projection directions resulting in best performance.

Acknowledgement. We would like to thank Huanxiang Lu, Ying Shi and Felix Schürmann from the Blue Brain Project for sharing the Brain data. We also thank Ying and Huanxiang for giving us an overview of the practical aspects of neuron delineation.

We thank Daniel Lebrecht and Anthony Holtmaat from the University of Geneva for sharing the Axons data.

MK would like to thank the European Commission for the support received from the FastProof PoC Grant.

AM acknowledges funding from the Swiss National Science Foundation.

References

Ahn, J., Kwak, S., 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation, in: CVPR, pp. 4981–4990.

Becker, C., Rigamonti, R., Lepetit, V., Fua, P., 2013. Supervised Feature Learning for Curvilinear Structure Segmentation, in: MICCAI, pp. 526–533.

Breitenreicher, D., Sofka, M., Britzen, S., Zhou, S., 2013. Hierarchical Discriminative Framework for Detecting Tubular Structures in 3D Images, in: MICCAI, pp. 328–340.

Briggman, K., Denk, W., Seung, S., Helmstaedter, M., Turaga, S., 2009. Maximin affinity learning of image segmentation, in: NIPS, pp. 1865–1873.

Bullitt, E., Zeng, D., Gerig, G., Aylward, S., Joshi, S., Smith, J., Lin, W., Ewend, M., 2005. Vessel Tortuosity and Brain Tumor Malignancy: A Blinded Study. Acad Radiol 12, 1232–1240.

Can, Y., Chaitanya, K., Mustafa, B., Koch, L., Konukoglu, E., Baumgartner, C., 2018. Learning to segment medical images with scribble-supervision alone, in: MICCAI, pp. 236–244.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S., Brox, T., Ronneberger, O., 2016. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, in: MICCAI, pp. 424–432.

Dai, J., He, K., Sun, J., 2015. Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation, in: ICCV, pp. 1635–1643.

Frangi, A., Niessen, W., Vincken, K., Viergever, M., 1998. Multiscale Vessel Enhancement Filtering. Lecture Notes in Computer Science 1496, 130–137.

Funke, J., Andres, D., Hamprecht, F.A., Cardona, A., Cook, M., 2012. Efficient Automatic 3D-Reconstruction of Branching Neurons from EM Data, in: CVPR, pp. 1004–1011.

Funke, J., Tschopp, F.D., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., Turaga, S.C., 2018. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. PAMI 41, 1669–1680.

Ganin, Y., Lempitsky, V., 2014. N4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms, in: ACCV, pp. 536–551.

Jing, L., Chen, Y., Tian, Y., 2018. Coarse-to-fine semantic segmentation from image-level labels. CoRR abs/1812.10885.

Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B., 2017. Simple does it: Weakly supervised instance and semantic segmentation, in: CVPR, pp. 1665–1674.

Kingma, D.P., Ba, J., 2015. Adam: A Method for Stochastic Optimisation, in: ICLR.

Koziński, M., Mosinska, A., Salzmann, M., Fua, P., 2018. Learning to Segment 3D Linear Structures Using Only 2D Annotations, in: MICCAI, pp. 283–291.

Kutulakos, K., Seitz, S., 2000. A Theory of Shape by Space Carving. IJCV 38, 197–216.

Law, M., Chung, A., 2008. Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux, in: ECCV, pp. 368–382.

Lin, D., Dai, J., Jia, J., He, K., Sun, J., 2016. Scribble-Sup: Scribble-Supervised Convolutional Networks for Semantic Segmentation, in: CVPR, pp. 3159–3167.

Maninis, K., Pont-Tuset, J., Arbeláez, P., Gool, L.V., 2016. Deep Retinal Image Understanding, in: MICCAI, pp. 140–148.

Markram, H., Muller, E., Ramaswamy, S., et al., 2015. Reconstruction and simulation of neocortical microcircuitry. Cell 163, 456–492.

Mattyus, G., Luo, W., Urtasun, R., 2017. Deep Roadmapper: Extracting Road Topology from Aerial Images, in: ICCV, pp. 3438–3446.

Meng, Q., Roth, H., Kitasaka, T., Oda, M., Ueno, J., Mori, K., 2017. Tracking and Segmentation of the Airways in Chest CT Using a Fully Convolutional Network, in: MICCAI, pp. 198–207.

Mnih, V., 2013. Machine Learning for Aerial Image Labeling. Ph.D. thesis. University of Toronto.

Mnih, V., Hinton, G., 2010. Learning to Detect Roads in High-Resolution Aerial Images, in: ECCV, pp. 210–223.

Mosinska, A., Marquez-neila, P., Kozinski, M., Fua, P., 2018. Beyond the Pixel-Wise Loss for Topology-Aware Delineation, in: CVPR, pp. 3136–3145.

Papandreou, G., Chen, L.C., Murphy, K., Yuille, A., 2015. Weakly-And Semi-Supervised Learning of a DCNN for Semantic Image Segmentation, in: ICCV, pp. 1742–1750.

Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., Zhou, Z., Gonzalez-Bellido, P., Oh, S., Chen, J., et al., 2014. Virtual Finger Boosts Three-Dimensional Imaging and Microsurgery as Well as Terabyte Volume Image Visualization and Analysis. Nature Communications 5, 4342–4355.

Peng, H., Zhou, Z., E.Meijering, T.Zhao, Ascoli, G., M.Hawrylycz, 2017. Automatic tracing of ultra-volumes of neuronal images. Nature Methods 14, 332–333.

Pinheiro, P., Collobert, R., 2015. From Image-Level to Pixel-Level Labeling with Convolutional Networks, in: CVPR, pp. 1713–1721.

Rajchl, M., Lee, M., Oktay, O., Kamnitsas, K., Passerat-Palmbach, J., Bai, W., Damodaram, M., Rutherford, M., Hajnal, J., Kainz, B., Rueckert, D., 2017. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. IEEE Trans. Med. Imaging 36, 674–683.

Rol, N., Timmer, E., Faes, T., Noordegraaf, A., Grünberg, K., Bogaard, H., Westerhof, N., 2017. Vascular narrowing in pulmonary arterial hypertension is heterogeneous: rethinking resistance. Physiological Reports 5.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation, in: MICCAI, pp. 234–241.

Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R., 1998. 3D Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. MIA 2, 143–168.

Shah, M., Merchant, S., Awate, S., 2018. Ms-net: Mixed-supervision fully-convolutional networks for full-resolution segmentation, in: MICCAI, pp. 379–387.

Shimoda, L., Laurie, S., 2013. Vascular remodeling in pulmonary hypertension. Journal of Molecular Medicine 91, 297–309.

Sironi, A., Turetken, E., Lepetit, V., Fua, P., 2016. Multiscale Centerline Detection. PAMI 38, 1327–1341.

Sweeney, M., Kisler, K., Montagne, A., Toga, A., Zlokovic, B., 2018. The role of brain vasculature in neurodegenerative disorders. Nature Neuroscience 21, 1318–1331.

Turetken, E., Becker, C., Glowacki, P., Benmansour, F., Fua, P., 2013. Detecting Irregular Curvilinear Structures in Gray Scale and Color Imagery Using Multi-Directional Oriented Flux, in: ICCV, pp. 1553–1560.

Vitanovski, D., Schaller, C., Hahn, D., Daum, V., Hornegger, J., 2009. 3D Annotation and Manipulation of Medical Anatomical Structures, in: Proceedings of SPIE on Medical Imaging, pp. 279–292.

Wegner, J., Montoya-Zegarra, J., Schindler, K., 2013. A Higher-Order CRF Model for Road Network Extraction, in: CVPR, pp. 1698–1705.

Zhao, Z., Yang, L., Zheng, H., Guldner, I., Zhang, S., Chend, D., 2018. Deep learning based instance segmentation in 3d biomedical images using weak annotation, in: MICCAI, pp. 352–360.