

# AI-Supported Segmentation of Industrial CT Data

Thomas Lang<sup>1,2</sup>, Tomas Sauer<sup>1,2</sup>

<sup>1</sup>FORWISS, University of Passau, Innstraße 43, 94032 Passau, Germany  
e-mail: thomas.lang@forwiss.uni-passau.de

<sup>2</sup>Fraunhofer EZRT, Flugplatzstraße 75, 90768 Fürth, Germany

## Abstract

The segmentation of datasets has always been a vital part of many image or volume processing applications, especially regarding tomography data. Common approaches nowadays use either manually tuned methods or rely on techniques like Deep Learning. However, such methods often are useful for segmenting only components of a certainly *well-defined* type which *mostly* suffices for clinical data but not for industrial applications anymore. In order to overcome these limitations we propose an interactive approach to volumetric segmentation encompassing robust classifiers and localized volume processing. The resulting algorithm is flexible enough to be used for a broad variety of different segmentation tasks while still generating high quality results. The local processing further enables the segmentation of larger volumes which cannot be handled by existing applications.

**Keywords:** Interactive Segmentation, Machine Learning, Big Volumes

## 1 Introduction

In the field of image processing for computed tomography (CT), segmentation is a crucial step in many applications. Segmentation means the partitioning of a volumetric dataset into non-overlapping regions. This either splits the entire dataset into different parts or only selected components are extracted from the (three-dimensional) image. We focus on the latter. Prominent examples include the detection and extraction of organs from human CT scans and subsequent analysis for diseases. The detection of the exact location of organs or other biological material is also very important for planning minimally invasive operations. In industrial computed tomography, the segmentation of objects is of interest too for, e.g., visual inspection of deformed components, comparisons with simulations or applications in dimensional metrology.

One of the major challenges in the industrial use case is the size of the scans: For a clinical dataset the size ranges typically from several hundred Megabytes up to few Gigabytes. For industrial CT the sizes can be significantly bigger due to higher resolutions and higher radiation levels, and with the ever-increasing effort in improving the scanners it is likely that they get continuously bigger. In this field, entire cars or small airplanes may be scanned at once [1], whose scan sizes range up from several hundred Gigabytes to even Terabytes. Yet another challenge is that there is a much higher variability of objects of interest in industrial scans compared to use cases where only a specific organ shall be segmented. Now, a multitude of different objects may be of interest for users which necessitates new and very general segmentation methods. Therefore, algorithms developed for clinical applications are unlikely to be applicable to such scans due to the large memory consumption problem not accounted for in clinical scenarios, and due to the large variety of different objects. Moreover, these algorithms may also explicitly depend on the object class like tumors which of course should not be searched for in, e.g., a scan of a car.

In order to overcome some of these problems, machine learning is applied to create robust segmentation models. However, classical algorithms like adaptive thresholding, cf. [2, 3], often do not produce good results due to imaging artifacts. Others, including works on atlas-based models [4, 5], graph-based methods [6, 7], evolving contours [8, 9] or neural networks [10–12] require either big training datasets, a lot of memory at once, or both. Since the sizes of CT scans are growing much faster than hardware capabilities and since annotated training data is hardly ever available, there is a trend towards interactive procedures, e.g., [13–15]. The main focus of such publications, however, lies on flexibility rather than the processing of large volumes. This paper introduces a segmentation procedure which combines interactive user input, local processing and machine learning methods. The interactivity allows to flexibly apply the same algorithm on a variety of different domains without the need to change the algorithm, while robust classifiers enable stable results and the purely local processing ensures that the method stays applicable to volumes without size restrictions. To the best of our knowledge, this work is one of the first to be explicitly designed for generality and processing large and *unique* computed tomography scans, i.e., in scenarios where no training data is available.

The organization of this paper is as follows: In Section 2, we will describe our proposed segmentation procedure and apply it on an example dataset to demonstrate the interactive improvement over several iterations. Section 3 will describe an experimental evaluation of our method on selected datasets and also qualitative results from different domains and Section 4 will conclude our contributions.

## 2 Method Description

In our approach, we apply pool-based active learning which is depicted in Figure 1. There an oracle, e.g., a human operator, is required to select a set of *seed voxels*, i.e., voxels which the oracle deems useful for distinguishing between the object that



shall be segmented and other regions. Additionally, the oracle needs to label them accordingly, i.e., either positively (belongs to the components under investigation) or negatively (does not belong to said components). In a next step, parameters can be set which include the environment size  $K_1 \in 2\mathbb{N} + 1$  and the set of features to compute. Alternatively, the most suitable subset of features could be automatically selected, e.g., by the method presented in [16]. Then, local features are computed. In detail, centered around each selected seed voxel a local environment of size  $K_1 \times K_1 \times K_1$  is extracted. From the voxels contained in the region features are extracted and concatenated into a feature vector. The feature set can be application-dependent, ours includes but is not limited to grayscale value statistics [17], a slightly modified version of the popular Local Binary Pattern [18] and inertia tensor features [19]. In practice we suggest that a domain expert selects a subset of such local features based on a priori knowledge. Using the overall set of obtained feature vectors, a model is trained with the goal of discriminating between the positive and negative feature instances. In our application, we chose a Support Vector Machine for this purpose since it is a binary classifier with beneficial properties including an inherent sparsity and the option of producing calibrated probability estimates as predictions. The training procedure also includes a K-fold cross validation step for hyperparameter tuning. After the training completed, the segmentation phase starts. There, we iterate in parallel over each voxel in the input volume and extract a local environment of a given size around it. For voxels where no full environment can be extracted, i.e., at the border regions of the volume, we set the value at these voxel positions in the output volume to zero. Given a local environment, we compute the same set of features and the trained model predicts a probability estimate indicating the likelihood that the currently processed voxel shall be segmented. This confidence value is assigned to the voxel position in the output volume. When the segmentation phase finishes, i.e., after one linear pass over the volume, the intermediate result can be evaluated, e.g., by means of visual inspection. If the result is satisfactory, the procedure ends. If not, the oracle can provide additional seed voxels which together with the previously selected ones form the new training dataset, and the method starts again.

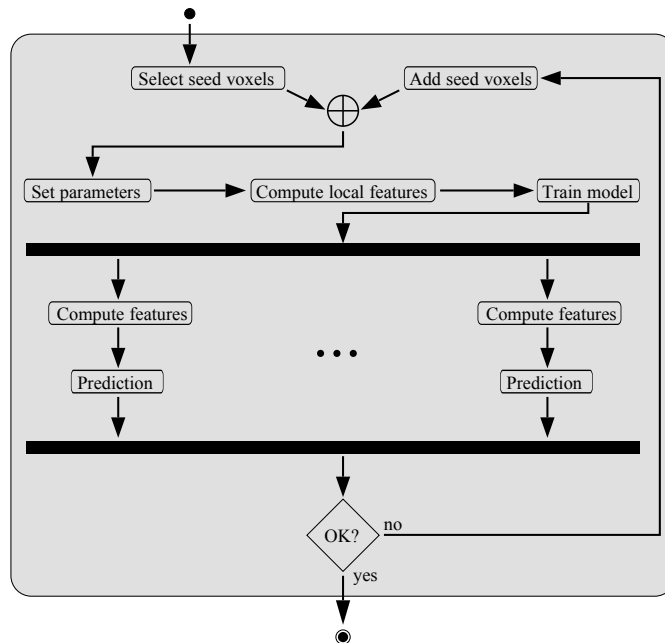


Figure 1: Active Learning for voxelwise segmentation

An open question is how to choose the samples the oracle shall label. A variety of methods exist, including *uncertainty sampling* or estimated error reduction [20]. However, given our use case of large industrial CT scans, only uncertainty sampling seems computationally feasible, i.e., we need to identify samples or regions the model is uncertain about. Fortunately, the probability estimates which are produced by the Support Vector Machine already hint the certainty of the model about each voxel. Furthermore, given a confidence value  $c \in [0, 1]$ , an uncertainty value  $u \in [0, 1]$  can be computed by considering that the critical confidence lies at  $1/2$ , i.e., when the SVM prediction lies close to the trained separating hyperplane. In our setting, we compute  $u = \exp(-g(|c - 1/2|))$  for some continuous and monotonically increasing function  $g: [0, 1/2] \rightarrow \mathbb{R}_{\geq 0}$  such that  $g(0) = 0$  and  $\lim_{x \rightarrow 1/2} g(x) = \infty$ .

An example of the interplay of interactive seed voxel selection, predicted confidences and uncertainties is depicted in Figure 2. There, we applied our interactive procedure to a Mahle motor piston dataset where our goal was to segment the steel ring from the aluminium piston body. Initially, we selected two seed voxels inside the ring regions and marked them positively, and two outside of it which are marked negatively. The selection is shown in form of blue diamonds. After applying the scheme described above, we obtain an output volume consisting of probability estimates which show that the model needs more samples as the confidences are higher outside the ring than in it. The uncertainty estimations also reflect this as they are quite high all over

the volume. In the next iteration (cf. Figure 2b) we selected additional seed voxels (blue diamonds) and combined them with the previous selection (red circles). The results improve considerably as shown by the confidences and uncertainties. The next two iterations stabilize the result and the uncertainty is nearly gone, thus we can stop labeling voxels at this point. Simple postprocessing on the confidences yields the desired result.

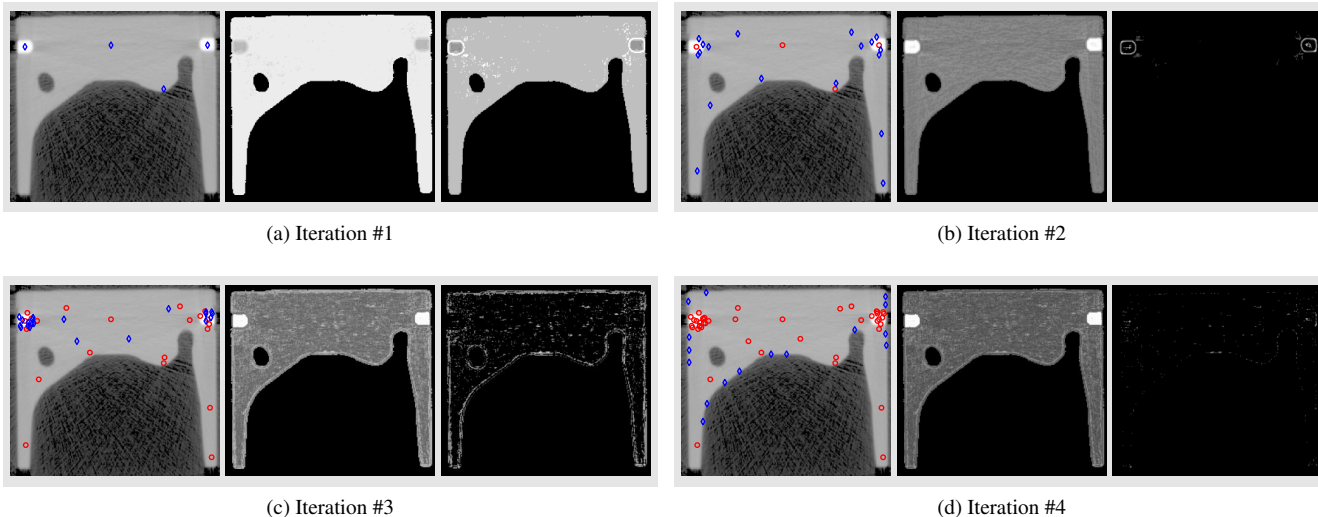


Figure 2: Interactive segmentation progression on a motor piston dataset, showing four iterations. Contrast was enhanced for better visibility, brighter shades indicate higher values. (a) shows, from left to right, the seed voxel selection, the predicted probability estimates and the uncertainty values computed from them. (b), (c) and (d) show the next three iterations, respectively.

### 3 Results

In this section, we will apply our interactive segmentation method to selected dataset and assess the segmentation performance both quantitatively and qualitatively. All datasets were generated by the Fraunhofer Development Center for X-ray Technology in Fürth and the data courtesy lies there as well. Exceptions to this are the scans labeled "Ring" and "Rumpf", which belong to a scan of a Komet Me-163 jet courtesy of Deutsches Museum München, Germany, and parts of a Peruvian mummy, courtesy of the Lindenmuseum in Stuttgart, Germany.

#### 3.1 Quantitative Results

For a quantitative evaluation, we selected seven datasets for which ground truth data was available. These are depicted in Figure 3 and Figure 5a. In case of the piston dataset, we aimed to segment the steel ring from the aluminium piston as in our example above. The same was attempted for the "Big Piston" scan consisting of two copies of the same piston and an additional Region of Interest of another scan. This scan was created with the intent of enlarging the volume size while also inserting components from a similar grayscale distribution so components are not only selected by their grayscale value. For the "Tasterwald" scan, the goal was to extract the small spheres while for "TPA" the shown part should be segmented in presence of high noise variance (not displayed for visualization purposes). Finally, for the "Ring" and "Rumpf" parts we aimed to segment the components overlaid in white. For each volume, we additionally selected a set of seed voxels just as we would interactively and fixed that set as well as a dataset-dependent set of features. Using this, we apply our proposed segmentation procedure, after which we applied speckle removal. That is, we iterate over the confidence value result and set the value of all voxels to zero which have less than  $\eta$  nonzero neighbors in a  $K_2 \times K_2 \times K_2$  environment centered around it. The used parameters are listed in Table 1. Finally, a connected components analysis groups voxels and the components overlapping with positively labeled voxels are extracted. These components then are evaluated against the ground truth data for the well-known performance metrics of Intersection over Union (IoU), precision, recall and the  $F_1$  score. The quantitative results are listed in Table 2. We see that, according to these metrics, good segmentation results can be obtained using our approach. At this point, we want to remind the reader that voxelwise comparison metrics are sensitive to individually misclassified voxels.

For another dataset we further measured the execution times of our approach which was implemented purely CPU-based inside a Fraunhofer framework and executed on a Dell Precision Tower 3620 having an Intel® Core™ processor with eight logical cores and an average clock rate of 4.2 GHz. The volume sizes herefore are specified in Table 3. The measurements are depicted in Figure 4a. Additionally, we also specified dataset-dependent thresholds below which all voxels are skipped and thus no classification computations need to be made there. That way, the processing can be accelerated significantly in interactive applications as is reflected in Figure 4b. In both images, the feature set encompassed grayscale value statistics and inertia

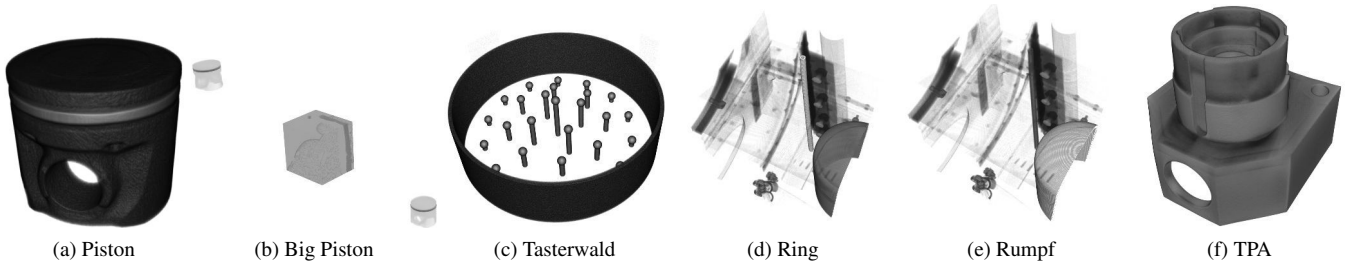


Figure 3: Datasets for which ground truth data is available. (a) shows a Mahle motor piston, (b) incorporates this piston in with a "Leberkasemmel" data. (c) shows a calibration dataset, while (d) and (e) show individual parts of a small airplane (courtesy of *Deutsches Museum*) where the components under investigation are highlighted in white, and (f) again is a calibration dataset.

Scan	$K_1$	#Seeds (+/-)	Speckle removal
Ford Fiesta	7	305 / 315	$K_2 = 5, \eta = 15$
Piston	5	48 / 114	$K_2 = 5, \eta = 18$
Big Piston	5	64 / 125	$K_2 = 5, \eta = 18$
Tasterwald	7	55 / 46	-
Ring	7	21 / 27	-
Rumpf	7	67 / 80	-
TPA	3	47 / 43	$K_2 = 3, \eta = 18$

Table 1: Experiment setup for quantitative evaluation.

Scan	IoU	Precision	Recall	$F_1$
Ford Fiesta	0.757	0.970	0.775	0.862
Piston	0.980	0.980	1.000	0.990
Big Piston	0.985	0.980	0.997	0.992
Tasterwald	0.883	0.999	0.884	0.934
Ring	0.950	0.965	0.984	0.974
Rumpf	0.949	0.963	0.985	0.974
TPA	0.989	0.992	0.997	0.994

Table 2: Quantitative evaluation results.

features [19], the runtime might increase or decrease depending on the computational complexity of the features as they have to be computed for every processed voxel environment. The bar plots per volume coarsely indicate the fact that our algorithm is linear with respect to the number of voxels. However, note that additional implementation details might influence the measurements including caching done by the operating system or scheduling operation for memory access. Overall, we achieve good runtime efficiency given the large dataset size on a purely CPU-based system.

### 3.2 Qualitative results

In addition to the quantitative evaluation, we now apply our method to unique scans where no training data is present. As before, we selected seed voxels interactively and defined a dataset-dependent skip threshold interactively as well. The configuration is listed in Table 4. For each result, we applied speckle removal and a connected component analysis as postprocessing. The results are visualized in Figure 5.

In contrast to segmenting the springs in the "Ford" scan, we now apply the *same* algorithm with a similar parametrization yet with different seed voxels on that scan in order to segment the windshield. The result shows that this thin planar structure can be detected well by our algorithm. In a similar fashion, we applied that method for segmenting a deformed part of a CT scan of a Honda Accord. Although that part has a very complex global structure, local patches can be successfully identified with each other and discriminated from others in the scan, thus allowing a good result. Outside of the industrial domain, we further used our interactive method to extract a wheat plant from a scan of the plant inside a pot filled with earth. The challenge there lies in the complex geometries of the roots and the low contrast between the roots and the surrounding earth. Even without dedicated models tracking root geometry, we were able to capture even most of the finer roots well by just interactively selecting voxels from and outside of them. In the final example, we used our proposed scheme to segment a scan of a Peruvian mummy into the ropes which hold it together and the result shows both the coarse ropes forming a net around the body and also the fine ones wrapped around its neck.

Scan	Dimensions	Size [GiB]	Skip threshold
Piston	$464 \times 464 \times 414$	0.166	7424
Wendler	$2073 \times 803 \times 616$	1.910	2560
BMW	$1195 \times 2372 \times 1008$	5.322	2304
Ford	$1178 \times 2446 \times 912$	4.895	15872
Mummy skull	$1504 \times 1771 \times 2513$	12.468	1280

Table 3: Runtime measurement configurations.

Scan	$K_1$	#Seeds (+/-)
Wheat	7	79 / 39
Ford	3	73 / 96
Honda	7	63 / 132
Mummy ropes	7	101 / 128

Table 4: Qualitative analysis configurations.

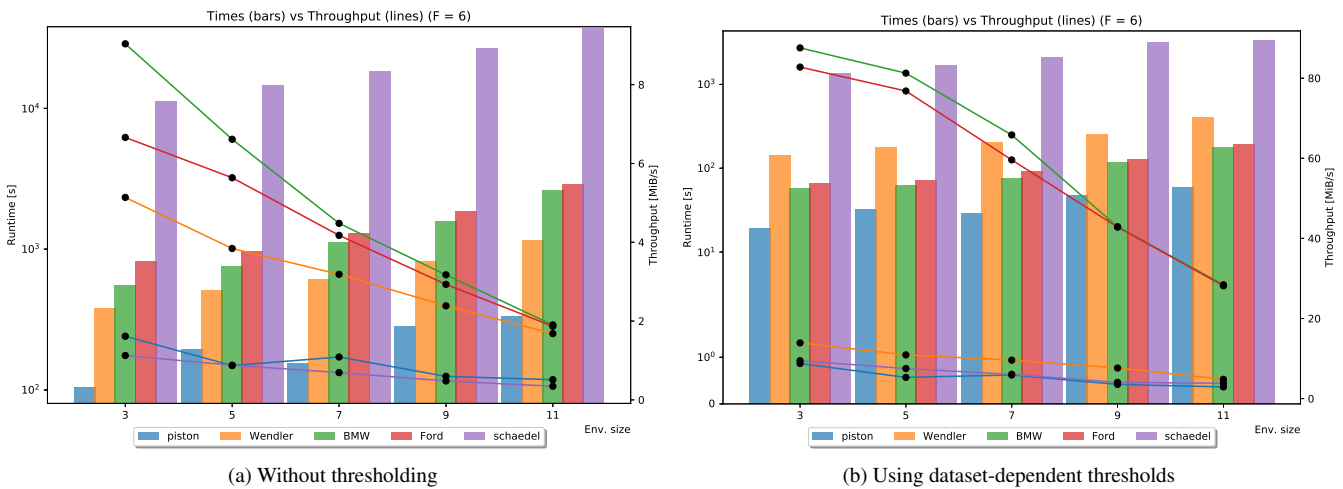


Figure 4: Runtime and throughput measurements for a subset of datasets for varying environment sizes  $K_1$ . The order of the bars is identical to the one in the legend, this view is best enjoyed in color. (a) shows the results when processing each individual voxel, while (b) shows the estimates when skipping irrelevant voxels according to the specified skip threshold.

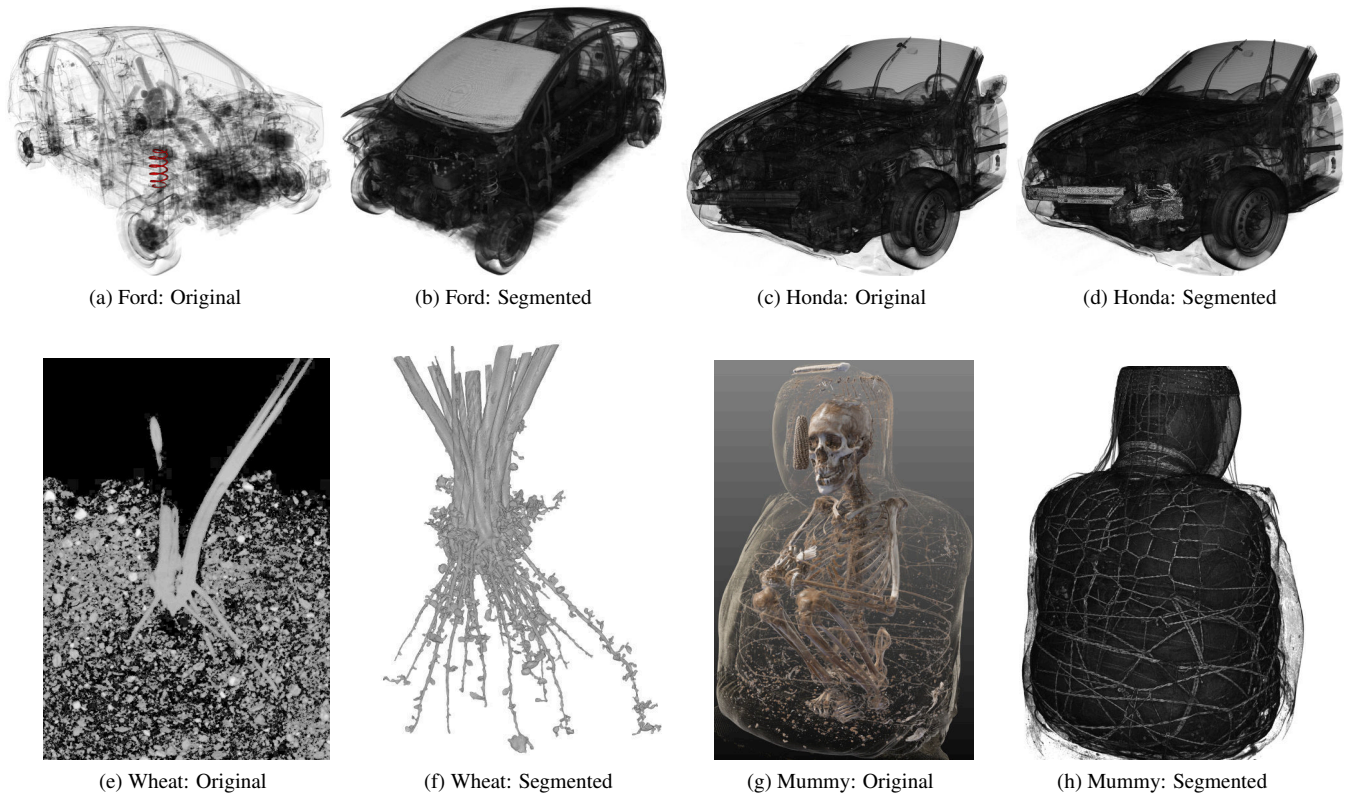


Figure 5: Qualitative segmentation results, each showing first a rendering of the original dataset and the segmentation result. The rendering in (g) was provided by Fraunhofer MEVIS. Contrast was enhanced where necessary. The results were overlaid over the original voxel dataset with the exception of the wheat roots for better visibility.

### 3.3 Comparisons to Related Works

In addition to our quantitative and qualitative results, we compared our algorithm to similar segmentation methods implemented in the freely available tools open\_iA, FIJI and Slicer3D. Specifically, we tried to segment the springs of the "Ford" scan but downsampled by a factor of two in each direction, thus processing a volume of roughly 700 Megabytes since none of these tools were able to process volumes of the original 5 Gigabytes size on the evaluation hardware described above.

With our method (in conjunction with postprocessing as described earlier), we were able to segment all four springs by just marking positive instances in **one** spring and some negative instances around it. The resulting evaluation yields a precision of

0.912, a recall of 0.893, a  $F_1$  score of 0.902 and an Intersection over Union of 0.822.

In the tool `open_iA`, we chose the probabilistic voxelwise segmentation via SVM. Their implementation uses the current voxel value exclusively and hyperparameters have to be set manually which is no easy task on its own. We varied the latter over a predefined set similar to the grid of parameters from which our hyperparameters are chosen automatically using a grid search approach. In addition, we used both the same seed voxel set as in our experiment and also newly chosen ones. Regardless of the setting combinations, our achieved result is an unstructured set of nonzero probability estimates located all over the surface of the object but not identifying any springs at all.

Next, we applied the region growing algorithm inside Slicer3D where we marked seeds inside the springs and outside with respective class labels. Contrary to our approach, a user needs to specify seeds for each spring, instead of finding all four with just information learned from one. The implementation itself has trouble processing the growing process for larger volumes, which is the primary reason we downsampled the input volume to 700 Megabytes in the first place. For this tool, we obtained a precision of 0.939, a recall of 0.699, a  $F_1$  score of 0.801 and an IoU of 0.668. These results are not bad but inferior to our segmentation performance in this case.

Finally, we used the SVM-based classifier inside FIJI which interprets the volume as a stack of images. The classifier allows for selecting a feature set interactively including grayscale statistics and, e.g., Hessian eigenvalue features. The major drawback of FIJI is that an image stack of the same size as the input volume is constructed in memory which is filled with feature vectors. Thus, a massive memory overhead is necessary and FIJI required 4.7 Gigabytes in main memory for a 78 Megabyte volume and our already downsampled Ford scan was not processable.

Overall, three popular and freely available software tools either are not well suited for segmenting industrial computed tomography data or have deficiencies when processing large voxel volumes.

## 4 Conclusion

This work introduced an interactive segmentation approach based on a Support Vector Machine. Its speciality is a purely local processing which enables the segmentation of volumes without size restrictions. Additionally, the interactivity allows passing the problem description and domain knowledge ad hoc into the system and thus the very same method can be applied to a variety of domains. A quantitative and qualitative evaluation demonstrated that very good results can be obtained while the user interaction stays intuitive and that repeated application of our algorithm can improve segmentation results continuously. We finally want to emphasize that our method is especially well suited to also segment scans for which no ground truth data exists, and the good results indicate that this method may very well be used for an efficient ground truth data generation requiring only minor manual correction.

## Acknowledgements

This work was supported by the project "Digitalisierung, Verarbeitung und Analyse kultureller und industrieller Objekte: Wertschöpfung aus großen Daten und Datenmengen - Big Picture" that has been promoted and funded by the Bavarian State Government under Grant No. AZ.: 43-6623/138/2.

We would like to thank two anonymous reviewers whose comments helped to improve the paper and clarify its main contributions.

## References

- [1] N. Reims, T. Schoen, M. Boehnel, F. Sukowski, M. Firsching, Strategies for Efficient Scanning and Reconstruction Methods on Very Large Objects with High Energy X-ray Computed Tomography, in: Proc. SPIE9212, Developments in X-Ray Tomography IX, 921209, 2014.
- [2] S. Zhou, Y. Cheng, Y. Wang, K. Dong, C. Guo, L. Bai, S. Tamura, Segmentation of the hip joint in CT volumes using adaptive thresholding classification and normal direction correction, *J. Chin. Inst. Eng.*, vol. 36, pp. 1059–1072, 2013.
- [3] C. Westin, A. Bhalerao, H. Knutsson, R. Kikinis, Using Local 3D Structure for Segmentation of Bone from Computer Tomography Images, in: Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., pp. 794–800, 1997.
- [4] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. Viergever, B. van Ginneken, Multi-Atlas-Based Segmentation With Local Decision Fusion - Application to Cardiac and Aortic Segmentation in CT Scans, *IEEE Trans. Med. Imaging*, vol. 28, no. 7, pp. 1000–1010, 2009.
- [5] T. Rohlfing, R. Brandt, R. Menzel, D. Russakoff, C. Maurer, Quo Vadis, Atlas-Based Segmentation?. In: Suri J.S., Wilson D.L., Laxminarayan S. (eds) *Handbook of Biomedical Image Analysis*, Topics in Biomedical Engineering International Book Series, Springer, Boston, MA, 2005.
- [6] B. Peng, L. Zhang, D. Zhang, A Survey of Graph Theoretical Approaches to Image Segmentation, *Pattern Recog.*, vol. 46, no. 3, pp. 1020–1038, 2013.
- [7] X. Chen and L. Pan, A Survey of Graph Cuts/Graph Search Based Medical Image Segmentation, in: *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 112-124, 2018, doi: 10.1109/RBME.2018.2798701.



- [8] D. Cremers, M. Rousson, R. Deriche, A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape, *Int. J. Comput. Vision*, vol. 72, pp. 195–215, 2007.
- [9] L. He, Z. Peng, B. Everding, X. Wang, C. Han, K. Weiss, W. Wee, A comparative study of deformable contour methods on medical image segmentation, *Image and Vision Computing*, vol. 26, pp. 141–163, 2008.
- [10] A. Taha, P. Lo, J. Li, T. Zhao, Kid-Net: Convolution Networks for Kidney Vessels Segmentation from CT-Volumes. In: Frangi A., Schnabel J., Davatzikos C., Alberola-López C., Fichtinger G. (eds) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, vol. 11073, Springer, Cham, 2018.
- [11] A. Hafiz, G. Bhat, A Survey on Instance Segmentation: State of the art, *Int. J. Multimed. Info. Retr.*, vol. 9, pp. 171–189, 2020.
- [12] F. Sultana, A. Sufian, P. Dutta, Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey, *Knowl.-Based Syst.*, vol. 201-202, pp. 106062, 2020.
- [13] M. Unger, T. Pock, W. Trobin, D. Cremers, H. Bischof, TVSeg - Interactive Total Variation Based Image Segmentation, in: *Proc. Brit. Mach. Vision Conf.*, 2008.
- [14] J. Santner, T. Pock, H. Bischof, Interactive Multi-label Segmentation. In: Kimmel R., Klette R., Sugimoto A. (eds) *Computer Vision – ACCV 2010*, Lecture Notes in Computer Science, vol 6492, Springer, Berlin, Heidelberg, 2010
- [15] F. Zhao, X. Xie, Interactive Segmentation of Medical Images: A Survey, *The 16th Conference on Medical Image Understanding and Analysis*, pp. 263–270 2012.
- [16] F. Song, Z. Guo and D. Mei, Feature Selection Using Principal Component Analysis, 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, 2010, pp. 27–30, doi: 10.1109/ICSEM.2010.14.
- [17] Pebay, Philippe Pierre. Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments, 2008. Web. doi:10.2172/1028931.
- [18] G. Zhao, M. Pietikainen, Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions, in: *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 06, pp. 915–928, 2007. doi: 10.1109/TPAMI.2007.1110
- [19] B. Jähne, *Spatial-Temporal Image Processing - Theory and Scientific Applications*, first ed., Springer, Berlin, 1993.
- [20] B. Settles, *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.