Automated Segmentation of Cell Organelles in volume electron microscopy using Deep Learning

Nebojša Nešić[1], Xavier Heiligenstein[2], Lydia Zopf[3], Valentin Blüml[3], Katharina S. Keuenhof[8], Michael Wagner[7], Johanna L. Höög[8], Heng Qi[4], Zhiyang Li[5], Georgios Tsaramirsis[6], Christopher J. Peddie[9], Miloš Stojmenović[1,*], and Andreas Walter[7,*]

[1]Department of Computer Science and Electrical Engineering, Singidunum University, Serbia
[2]CryoCapCell, France
[3]BioImaging Austria, Austria
[4]Department of Computer Science, Dalian University of Technology, China
[5]Department of Computer Science, Dalian Maritime University, China
[6]Higher Colleges of Technology, United Arab Emirates
[7]Aalen University of Applied Sciences, Germany
[8]Department for Chemistry and Molecular Biology, University of Gothenburg, Sweden
[9]Electron Microscopy STP, The Francis Crick Institute, United Kingdom
[*]Corresponding authors: Andreas Walter, Andreas.walter@hs-aalen.de; Miloš Stojmenović, mstojmenovic@singidunum.ac.rs

November 11, 2023

Abstract

Recent advances in computing power triggered the use of Artificial Intelligence in image analysis in life sciences. To train these algorithms, a large enough set of certified labelled data is required. The trained neural network is then capable of producing accurate instance segmentation results, that will then need to be re-assembled into the original dataset: the entire process requires substantial expertise and time to achieve quantifiable results. To speed-up the process, from cell organelle detection to quantification across modalities, we propose a deep learning based

approach for Fast AutoMatic Outline Segmentation (FAMOUS), that involves organelle detection combined with image morphology, and 3D meshing to automatically segment, visualize and quantify cell organelles within volume electron microscopy datasets. From start to finish, FAMOUS provides full segmentation results within a week on previously unseen datasets. FAMOUS was showcased on a dataset acquired using a focused ion beam scanning electron microscope (FIBSEM), and on yeast cells acquired by transmission electron microscopy.

Research highlights

Introducing a rapid, multimodal machine-learning workflow for 3D cell organelle segmentation. Applied successfully to diverse datasets and cell lines, it outperforms manual methods, enabling high-throughput quantitative cell biology.

Introduction

Imaging in Life Sciences is currently experiencing a boost, and imaging data are growing exponentially. Biological processes, ultrastructure and molecules can now be visualized at unprecedented resolution in time, depth and scale [25] [26]. Large volumetric reconstructions of entire cells can be routinely achieved at nanometer resolution using volume electron microscopy (vEM). Quantitative analysis of such large amounts of data is the novel bottleneck in biological projects. Within a decade, what used to be considered as extreme large datasets [14] and analyzed over a PhD period, is now routinely processed (12Gb RAM is common on laptops). One important goal in vEM is to quantitatively annotate and segment the volume stacks to quantify organelle distributions and shapes to understand the structure-function relationship. Many diseases are associated with abnormal organelle morphologies and distributions within cells, including a growing number of neurodegenerative diseases, such as Alzheimer's [28] or Lewy- Body-Dementia [10]. EM visualizes ultrastructural details and rich contextual information based on protein/lipid or stain-density gradients. Not only the structures of interest are visible, but also all membrane-delineated ultrastructural cell content. The signal-to-noise ratio is low and, up to date, at the expense of time, organelles have mainly been deciphered from one another based on their membrane delineation by the human eye. As conventional segmentation schemes are often based on thresholds or manipulations of the image histogram assuming that strong gradients match object boundaries, unsupervised binarization algorithms, such as minimum error thresholding, maximum entropy thresholding or Otsu's single-level method [18], fail to reliably identify and segment organelles. In practice, automatic segmentations generated based on thresholds or manipulations of the image histogram usually require extensive manual post-editing to achieve the desired accuracy. Therefore, segmentation of cell organelles is currently mainly performed manually using segmentation tools included in commercial software, such as AMIRA [23] or Imaris [3], or freeware

tools, such as ImageJ/Fiji [21] [20], IMOD [15], or Ilastik [5]. For a whole HeLa cell imaged at 5 nm isovoxel resolution using a FIBSEM setup at 5nm isovoxel resolution, manual segmentation of important organelles (such as mitochondria, nucleus, ER or endosomes) will take several months if carried out by a single person and requires comparative segmentation to cross-validate the results.

Progresses in computational methods for automatic segmentation of organelles in vEM has led to increasingly accurate results [22], using for example training of classifiers to detect supervoxels that most likely belong to the boundary of the segmentation target [16]. While there are packages available that already use learning-based approaches, such as Ilasitk or Cell Profiler, they usually do not allow training on new datasets limiting their application to a specific and small range of datasets or require substantial expertise in image analysis.

For light-microscopy datasets (acute signal-to-noise ratio), several deep-learning solutions for segmentation and quantification, such as cell detection or morphological measurements, have already been published [19], [13], [7]. Object detection is a technique that allows the computer to find the location (x and y coordinates, width and height) of a particular shape, or organelle in an image. Instance segmentation takes this one step further and isolates the foreground pixels of the shape or organelle. U-Net [19] was pioneering work in the field of instance segmentation that was initially applied to microscopy data. The U-shaped Deep Learning architecture is capable of capturing and generalizing high level descriptors of image data as the information reaches the convolutional valley of the U. By concatenating this encoded data with the finer convolutional layers from higher levels, the network can reconstruct the boundary of the shape instance. The U- Net architecture is used as the backbone of Etch a cell [2], a crowd- sourced approach to generate large quantities of labelled data.

To the best of our knowledge, for comprehensive segmentation of all organelles in large volumetric EM data sets, only a few open-access approaches have been suggested. The trainable WEKA segmentation toolkit [4] can train segmentation pipelines using generic hand-tailored image features. DeepEM3D [27] aims at improving reproducibility while providing open access to deep-learning algorithms for image segmentations using a cloud-based setup that does not require a local GPU. Other approaches focus on single imaging modalities, such as COSEM for automated identification of all intracellular substructures within isotropic FIB/SEM datasets, or on specific organelle tools for semi-automatic 3D segmentation, including mitochondria or neuron tracing [12]. Last but not least, Ilastik 1.3.3 contains modules for pixel classification via training using simple brush strokes. This approach is designed for users without machine learning expertise, and may prove useful in simple segmentation scenarios where optimizing the training parameters yields little benefit. The very first commercial solutions have also been launched [1] and rely on a large internal human expertise of the segmenting scientist to edit the final model.

In summary, despite the urgent need in the vEM and structural biology communities, there is no quantitative segmentation workflow available that was proven successful for different biological single cells across volume EM modali-

ties. To improve the quantitative performance of automated image segmentation of large volumetric datasets, we identify the need for a generic, accessible and tractable segmentation software that is assesed against the current gold standard of manual segmentation. YOLO [6] is a 'you only look once' framework for deep learning that accurately performs image based object detection in real time with minimal training data. It re-frames the object detection problem so that the model not only infers the category of the object, but also its position and size in the image at the same time.

We present our machine-learning pipeline and algorithm for automated segmentation of organelles. We showcase the workflow for two different vEM approaches: FIBSEM of a HeLA cell and an array tomography of yeast cells using TEM and quantitatively compare the results with manually segmented datasets as the current gold standard. Since it does not make any a priori assumptions about the morphology of the organelles to be segmented, the pipeline can be easily applied to segment diverse organelles across cell types and modalities, including Soft X-ray microscopy [25]. FAMOUS, although perfectible, yields to a comparable accuracy in classification and localization to manually segmented dataset, within a fraction of the period.

Results

The amount of data generated in vEM for life sciences usually ranges from gigabytes to terrabytes per dataset. It is practically impossible to manually segment out the information content of a vEM dataset in the reasonable time period of a publication, let alone to create meaningful statistics across cells. To automate image segmentation, we propose a simplified pipeline where we exploit innovative image analysis based on neural networks to deliver a full volume segmentation of cell organelles within a week.

First all structures of interest within a limited subset of the data, i.e. from about only 1% of the entire 3D stack, need to be accurately flagged. This annotation is used to train the image recognition algorithm, isolate the structures of interest, run the image-processing pipeline and feedback the resulting outlined structures into the 3D rendering software Blender [8] where the scientific analysis can be initiate her/his scientific analysis. Upon completion of the segmentation, a 3D dataset is recieved, composed of image stacks and all the organelles segmented and organized in groups. Singular organelles are unique objects and are grouped together into coherent classes, allowing arbitrary subset creation and visualization to focus on. (Figure 1)

Detection and classification performance evaluation

To evaluate the performance of our automated segmentation pipeline (denoted as stack F (green)), the dataset was segmented twice manually by two independent experts (denoted as M1(red) and M2(blue)). Comparative studies were conducted between the manually segmented stacks and the manual and the au-

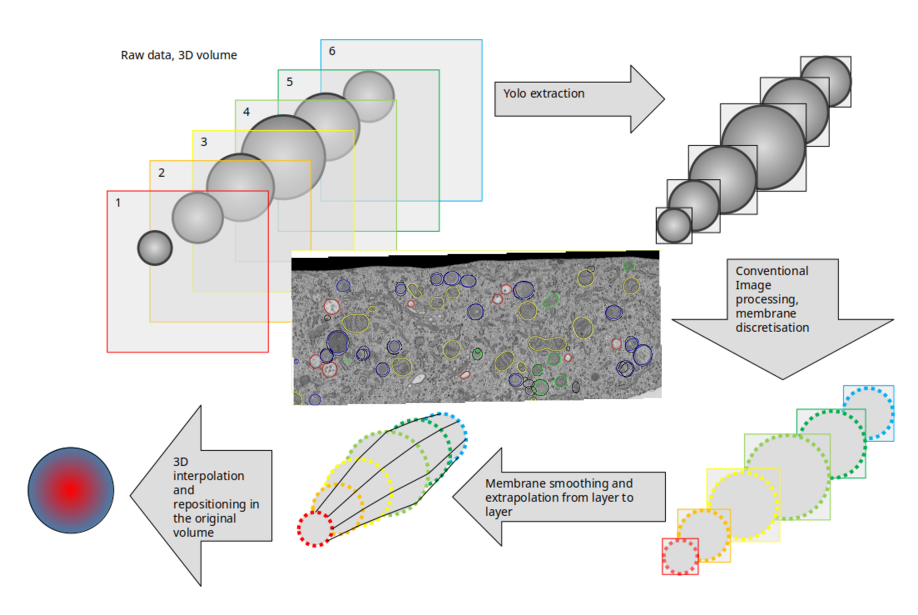


Figure 1: The workflow from the developers point of view

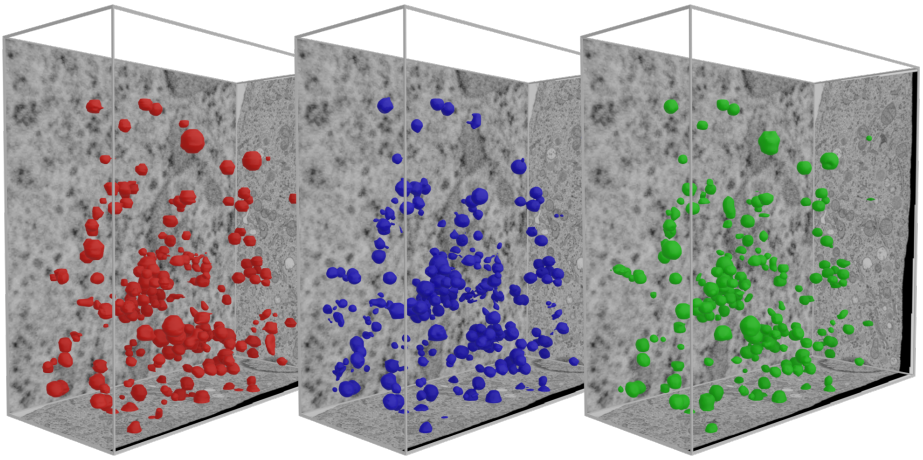


Figure 2: FIBSEM final visualization - Early Endosomes - M1(red) - M2(blue) - Automatic(green)

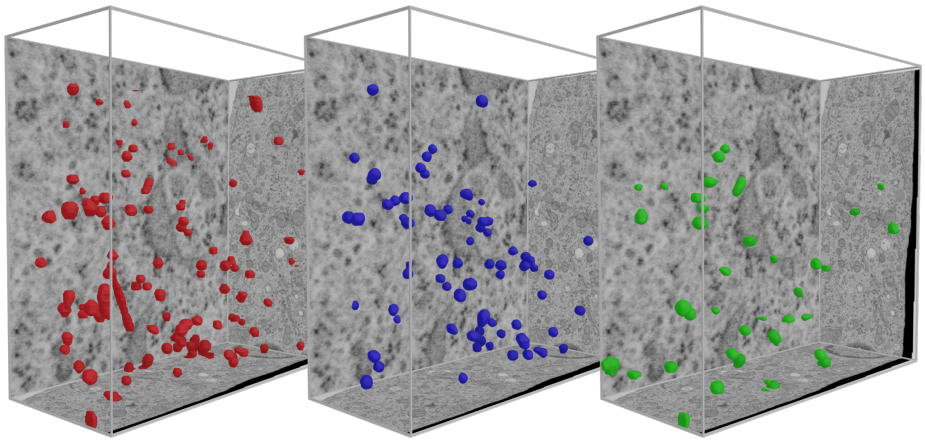


Figure 3: FIBSEM final visualization - Late Endosomes - M1(red) - M2(blue) - Automatic(green)

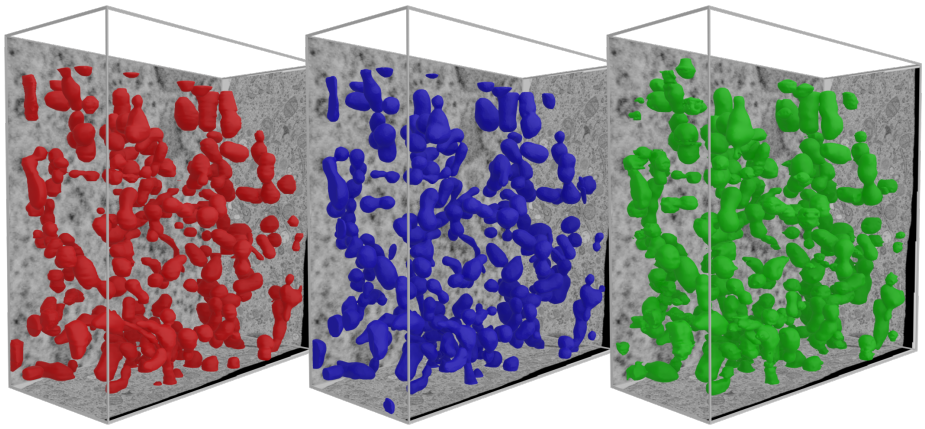


Figure 4: FIBSEM final visualization - Mitochondria - M1(red) - M2(blue) - Automatic(green)

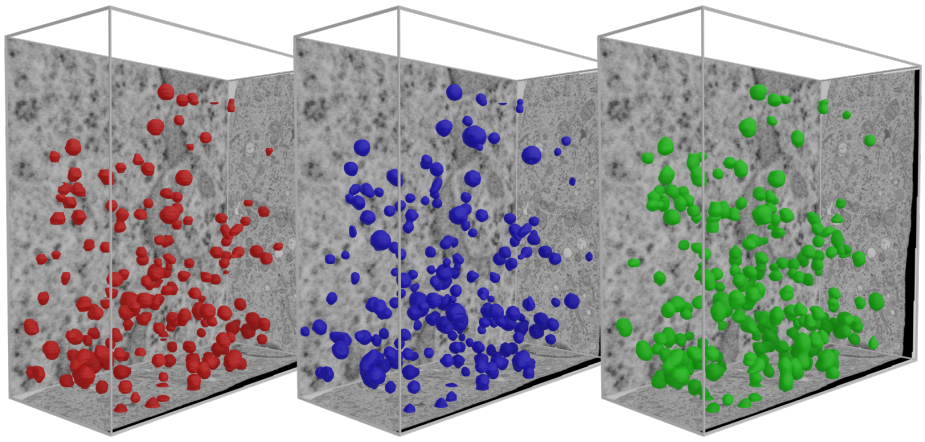


Figure 5: FIBSEM final visualization - Lysosomes - M1(red) - M2(blue) - Automatic(green)

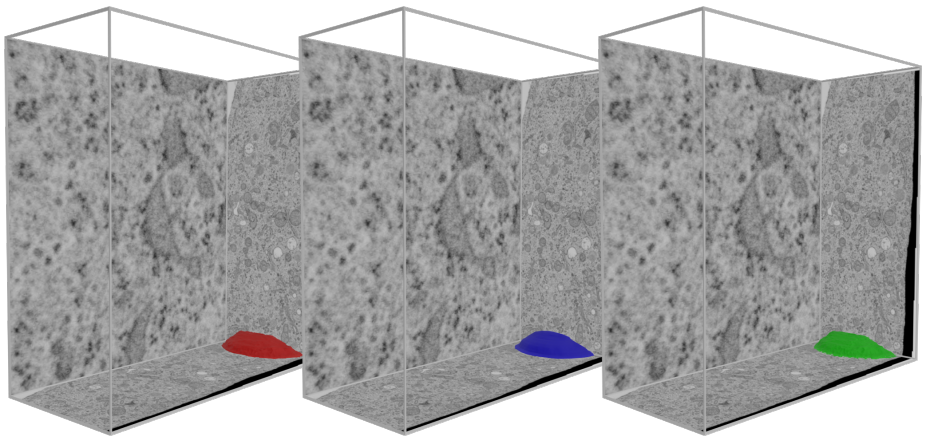


Figure 6: FIBSEM final visualization - Nucleus - M1(red) - M2(blue) - Automatic(green)

tomatically segmented stacks. This gave us insights into the deviation between two manual segmentations and served to benchmark the automatic segmentation. The segmented organelles, in the FIBSEM dataset, were early endosomes, late endosomes, mitochondria, lysosomes and the nucleus. The difference in the numbers of detected organelles was quantified for each organelle category, and each diverging label or misdetection was identified and analyzed further. Taking both manual workflows as the ground truth, and the FAMOUS detection as the comparison, we classified all organelles into:

- Objects correctly identified by FAMOUS are considered True Positives
- Object inadequately identified by FAMOUS are False Positives
- Object identified in the manual workflow and not identified by FAMOUS are False Negatives
- Object detected by FAMOUS and missed by the manual workflow are true negatives

In a few cases, FAMOUS wrongly identified one object as multiple objects that share the same space, the TP and FP values were adjusted accordingly, to avoid getting multiple positive identifications of the same object. To compare the performance of FAMOUS on the macroscopic level (detection efficiency, identification and classification performance), we used four separate criterions:

- Precision - of all the classes how many were correctly predicted. Qualified as Precision $= TP/(TP + FP)$
- Sensitivity - if a positive rate is predicted how often does this take place? Qualified as Recall $= TP/(TP + FN)$
- The harmonic mean of Precision and Recall. Qualified as F1 $= 2 * TP/(2 * TP + FP + FN)$
- The similarity between the manual and automatic segmentation. Qualified as Jaccard index as $TP/(FP + TP + FN)$

Comparison of volumes, areas & evaluation metrics

After the identification and classification, the organelles were segmented by applying conventional histogram-based filters to a cropped-out region and averaging noise out. This computationally efficient pipeline uses parallel processing (GPU) on each cropped-out region. No large computing capacity is required.

To evaluate our segmentation approach, we conducted a volume comparison of each individual class (Figure 18). The total volume of all objects in an individual class was calculated for both the manual segmentations and the automatic workflows and plotted to quantify differences at the whole volume scale.

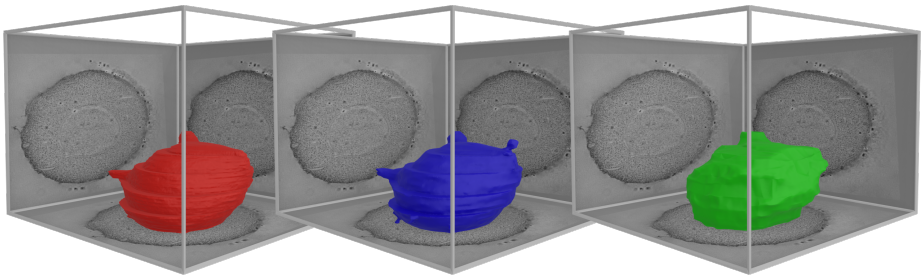


Figure 7: Yeast final visualization - Nucleus - M1(red) - M2(blue) - Automatic(green)

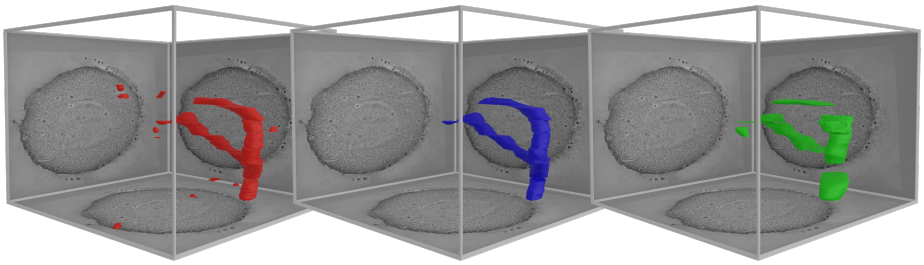


Figure 8: Yeast final visualization - Mitochondria - M1(red) - M2(blue) - Automatic(green)

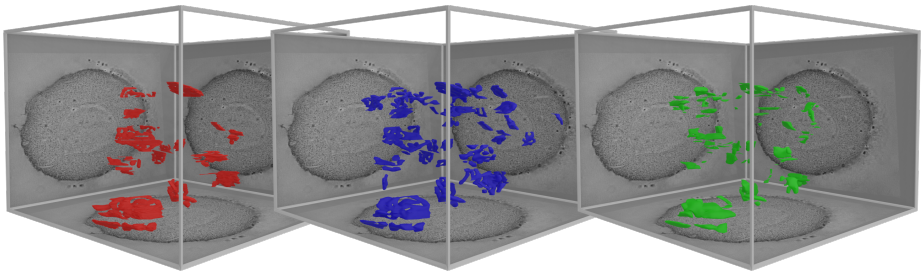


Figure 9: Yeast final visualization - Golgi - M1(red) - M2(blue) - Automatic(green)

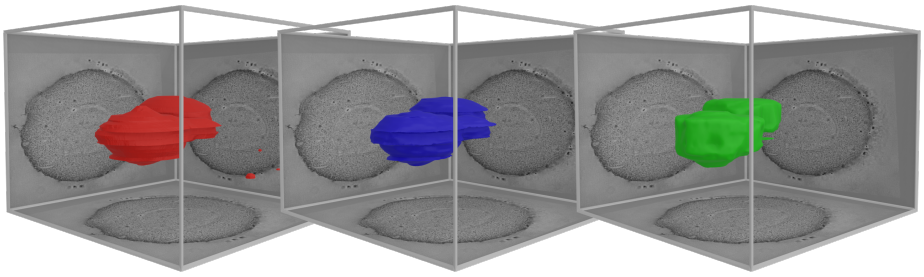


Figure 10: Yeast final visualization - Vacuole - M1(red) - M2(blue) - Automatic(green)

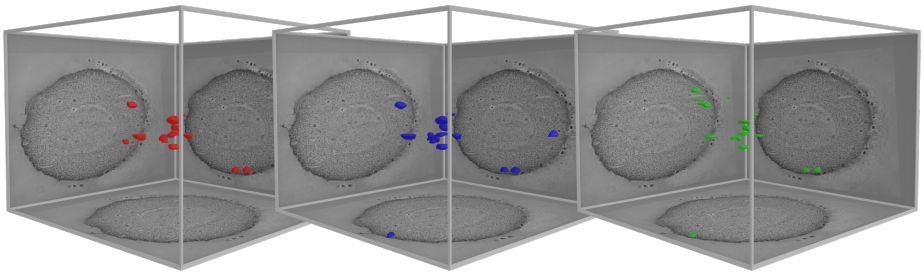


Figure 11: Yeast final visualization - Multivesicular bodies - M1(red) - M2(blue) - Automatic(green)

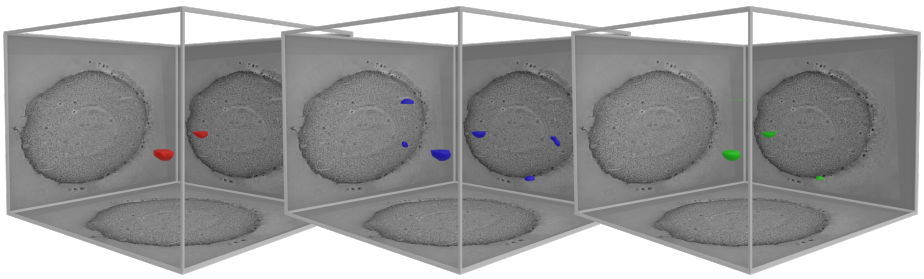


Figure 12: Yeast final visualization - Lipid droplets - M1(red) - M2(blue) - Automatic(green)

Table 1: Comparing Volume overlap between manual (M1, M2) and automatic segmentation
- FIBSEM

Class	Automatic-M1	Automatic-M2	M1-M2
Early Endosomes	65.04%	71.75%	86.31%
Late Endosomes	31.08%	9.95%	74.77%
Mitochondria	95.20%	94.94%	90.99%
Lysosomes	93.60%	69.91%	71.71%
Nucleus	98.31%	99.01%	99.54%

We then explored the intersection value, i.e. how much one unique object differs in its segmented properties (surface, periphery, center of mass etc) from one method to the other and what is the distribution amongst that class. This was achieved using a Boolean union operator, which joins two objects into one, while removing their intersection. The volume of the automatic workflow was subtracted from the total volume of both the automatic and manual workflows thus providing the difference between the two workflow volumes. The volumes were calculated in μm^3(Table 1). The volume results are dependent on the correct classification of objects into their classes and the position of the misclassified objects. As was expected from the previous metrics, there is a very good volume overlap between all automatic and manually segmented organelles, which is in the range of that between the two manually segmented datasets. Only the late endosomes were not faithfully assigned. Late endosomes are volumetrically the smallest class, and only a few misclassified organelles can create a large distortion in the total volume of the entire class, thus skewing the final numbers.

The total volume distribution of the dataset is presented in in figures 13, 14, 15, 16 and 17. As can be appreciated in Tables 2 and 3, the overall precision, sensitivity and Jaccard Indices achieved by FAMOUS are comparable with those achieved between the two gold standards of experienced manual segmentators (Table 4). While the manual segmentation for the entire FIBSEM dataset was achieved by each segmentator in about 200 hours and that of the array tomography dataset within 120 hours, including visualization, our presented automated segmentation pipeline required 12 and 8 hours, respectively, in terms of actual (guided) input time by the user, including the preparation of a training set. Within a about a sixteenth of time, FAMOUS reliably automated a full-stack segmentation, visualization and quantification of an entire cell acquired by vEM - with an accuracy similar to the current gold standard. The workflow thus substantially facilitates quantification and analysis in high-resolution structural biology and can be quickly reproduced as described in the Methods section. A similar statistical analysis was done on the yeast dataset as well, and can be seen in tables 5, 6 and 7.

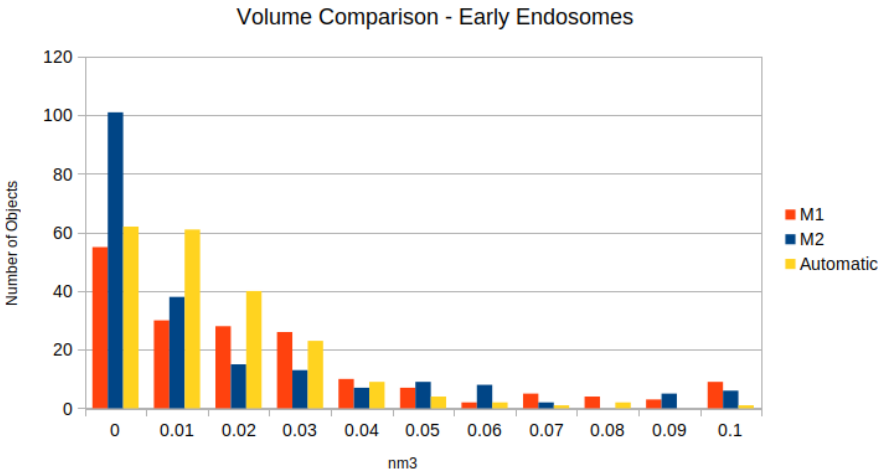


Figure 13: Volume comparison of the FIBSEM dataset - Early Endosomes

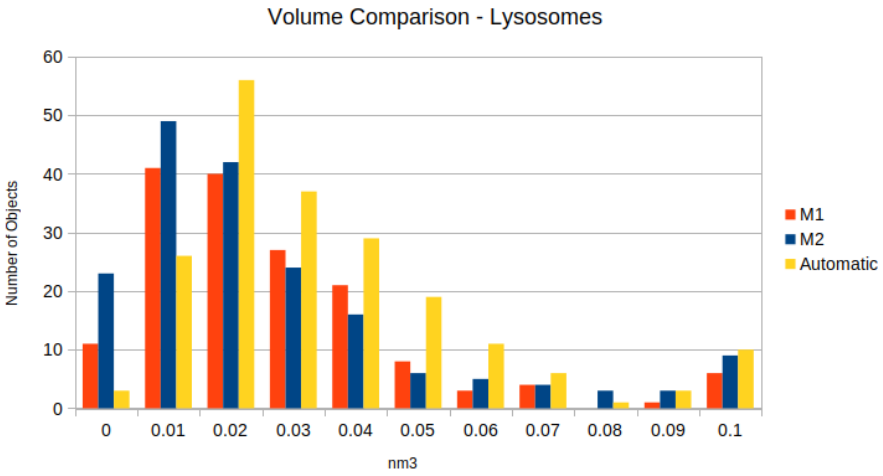


Figure 14: Volume comparison of the FIBSEM dataset - Lysosomes

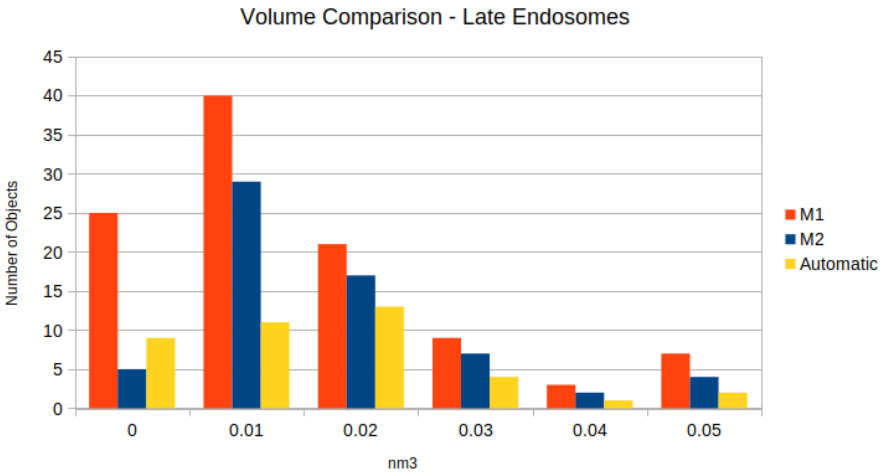


Figure 15: Volume comparison of the FIBSEM dataset - Late Endosomes

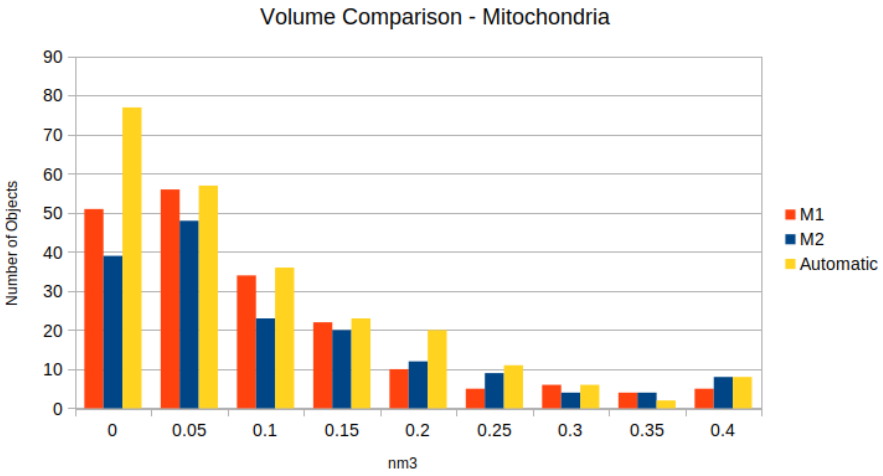


Figure 16: Volume comparison of the FIBSEM dataset - Mitochondria

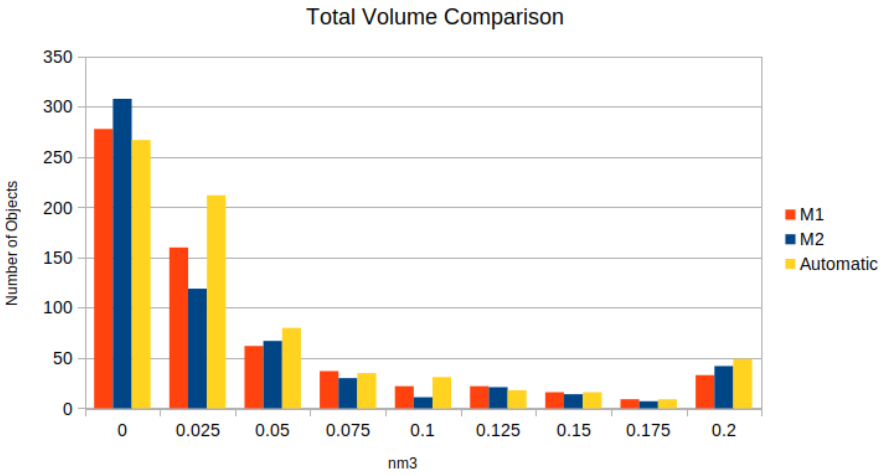


Figure 17: Volume comparison of the FIBSEM dataset - Total

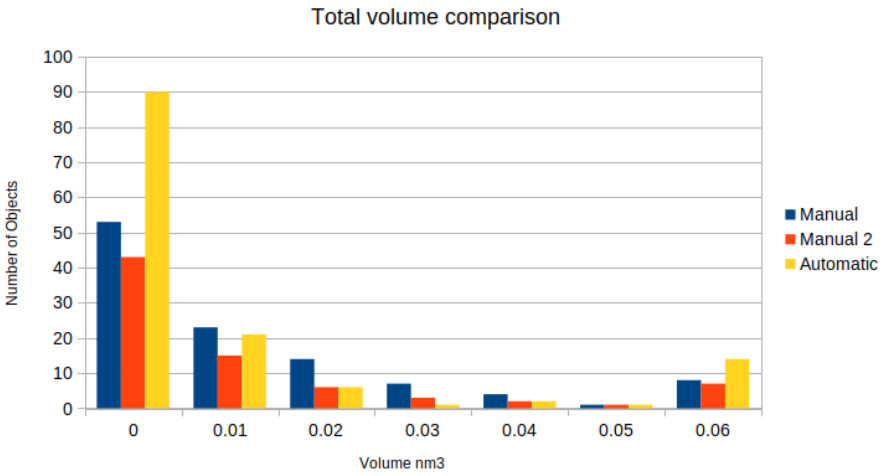


Figure 18: Volume comparison of the Yeast dataset - Total

Table 2: Precision, Recall, Jaccard index and F1 scores of the automatically segmented data by FAMOUS compared to M1 - FIBSEM

Class	Precision	Recall	Jaccard Index	F1 Score
Early Endosomes	84.02%	79.33%	68.93%	81.61%
Late Endosomes	37.50%	14.29%	11.54%	20.69%
Mitochondria	84.43%	92.27%	78.85%	88.18%
Lysosomes	72.83%	82.72%	63.21%	77.46%
Nucleus	100.00%	100.00%	100.00%	100.00%

Table 3: Precision, Recall, Jaccard index and F1 scores of the automatically segmented data by FAMOUS compared to M2 - FIBSEM

Class	Precision	Recall	Jaccard Index	F1 Score
Early Endosomes	82.42%	66.67%	58.37%	73.71%
Late Endosomes	17.50%	11.29%	7.37%	13.73%
Mitochondria	88.57%	92.81%	82.89%	90.64%
Lysosomes	74.19%	75.00%	59.48%	74.59%
Nucleus	100.00%	100.00%	100.00%	100.00%

Table 4: Precision, Recall, Jaccard index and F1 scores of the data - M1 vs M2 - FIBSEM

Class	Precision	Recall	Jaccard Index	F1 Score
Early Endosomes	85.47%	75.00%	66.52%	79.90%
Late Endosomes	39.05%	64.06%	32.04%	48.52%
Mitochondria	79.38%	92.22%	74.40%	85.32%
Lysosomes	88.89%	78.26%	71.29%	83.24%
Nucleus	100.00%	100.00%	100.00%	100.00%

Table 5: Precision, Recall, Jaccard index and F1 scores of the data - M1 - yeast

Class	Precision	Recall	Jaccard Index	F1 Score
Cell	100.00%	100.00%	100.00%	100.00%
Nucleus	100.00%	100.00%	100.00%	100.00%
Mitochondria	100.00%	100.00%	100.00%	100.00%
Golgi	63.64%	75.90%	52.94%	69.23%
Vacuoles	100.00%	100.00%	100.00%	100.00%
Multivesicular bodies	62.50%	93.75%	70.00%	75.00%
Lipid droplets	37.50%	50.00%	27.27%	42.86%

Table 6: Precision, Recall, Jaccard index and F1 scores of the data - M2 - yeast

Class	Precision	Recall	Jaccard Index	F1 Score
Cell	100.00%	100.00%	100.00%	100.00%
Nucleus	100.00%	100.00%	100.00%	100.00%
Mitochondria	100.00%	100.00%	100.00%	100.00%
Golgi	46.46%	100.00%	46.46%	63.45%
Vacuoles	100.00%	100.00%	100.00%	100.00%
Multivesicular bodies	47.83%	84.62%	44.00%	61.11%
Lipid droplets	25.00%	100.00%	25.00%	40.00%

Table 7: Precision, Recall, Jaccard index and F1 scores of the data - M1 vs M2 - yeast

Class	Precision	Recall	Jaccard Index	F1 Score
Cell	100.00%	100.00%	100.00%	100.00%
Nucleus	100.00%	100.00%	100.00%	100.00%
Mitochondria	100.00%	100.00%	100.00%	100.00%
Golgi	95.65%	53.01%	51.76%	68.22%
Vacuoles	100.00%	100.00%	100.00%	100.00%
Multivesicular bodies	92.31%	75.00%	70.59%	82.86%
Lipid droplets	100.00%	33.33%	33.33%	50.00%

Discussion

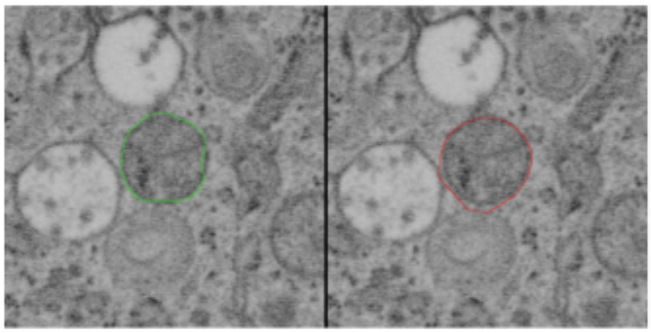
In this paper, we have presented a novel automatic segmentation tool for vEM datasets across modalities that segements cell organelles as reliably as manual segmentation by visual inspection, as quantified by Jaccard Indices and volume comparisons. The workflow (FAMOUS) can analyse and quantify an entire

dataset of several terabytes within a few hours, i.e. in a fraction of time compared to manual segementation. FAMOUS will hence significantly contribute to high throughput and automation in vEM, and help to push the field towards quantitative imaging and statistically solid results.

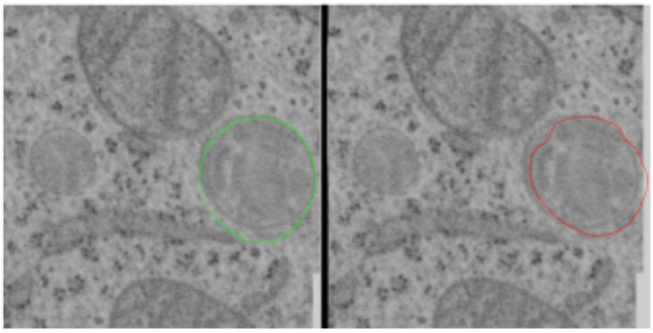
An important issue that arose early is that the manual segmentation cannot be viewed as fully and exclusively representative of the actual ground truth data, mainly due to human error. There were instances where the automatic segmentation identified organelles accurately, but the manual segmentation did not classify the organelles in the same class as the automatic or missed them entirely (Figure 20). In such cases, the automatically segmented organelle was labeled as an error. These cases biased the final accuracy numbers of the automatic segmentation, and can only be corrected by visual inspection. The subjective assessment of the expert who carries out the manual segmentation plays a significant role in the final results, meaning that different experts classify the same organelle into different classes, as quantified by the Jaccard Indices below 1 between the two manually segmented datasets. For a better illustration of such cases, a 3D mesh intersection with the slice was done, after which an outline of the intersection was created. The automatically segmented outline is shown on the left in green, while the manually segmented outline is shown on the right in red (Figure 19). In addition, the automatic workflow identified organelles that the manual segmentation did not (Figure 20). The reverse situation is also present, where the automatic workflow failed to identify organelles that the manual did. However, in these cases, the automatic workflow did not fail in recognizing that the organelle existed, but the organelle was identified as the wrong class. This issue only arises when two classes have similar visual features. In the FIBSEM dataset, the organelles that fall into this category are the late endosome and lysosomes.

While the manual segmentation comparison shows better number for the late endosome class, when compared with the other classes, late endosomes are shown to be the most problematic there as well. For the FIBSEM dataset, the automatic segmentation outperforms the manual segmentations in the detection of the early endosomes and mitochondria, and, as stated above, slightly underperforms in the detection of late endosomes and lysosomes. For the array tomography yeast dataset, FAMOUS and the manual segmentation yield similar accuracy in the detection and segmentation of the organelles (compare Tables 2 and 3), when comparing the mean Jaccard indices for both manual and automatic segmentations. We observe that our segmentation strategy does not overestimate the organelles in comparison to the manual segmentation.It is interesting to note that even experienced scientists cannot unambiguously agree upon assigning organelle structures in a cell volume, which provides another argument on why automation of the process (and hence objectifying it) is of utmost importance.

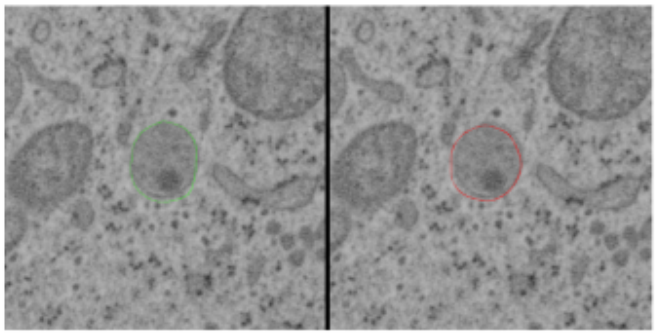
FAMOUS only struggled with complex objects that were connected by small "bridges" between the larger, more rounded parts of the object (Figure 21). In these cases, the automatic segmentation sometimes identified every major part of the complex object as a separate entity and did not recognise them as a



Automatic: Late Endosome; Manual: Mitochondria
Correct: Late Endosome

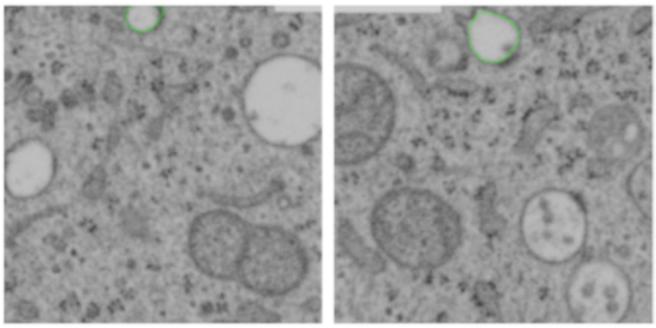


Automatic: Lysosome; Manual: Late Endosome
Correct: Lysosome

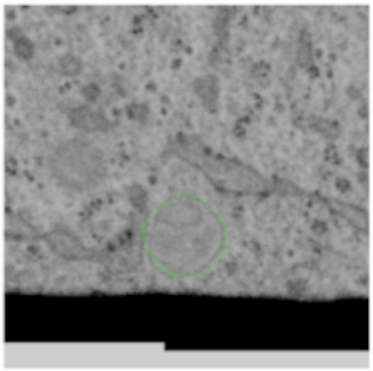


Automatic: Late Endosome: Manual: Lysosome
Correct: Late Endosome

Figure 19: Manual segmentation errors

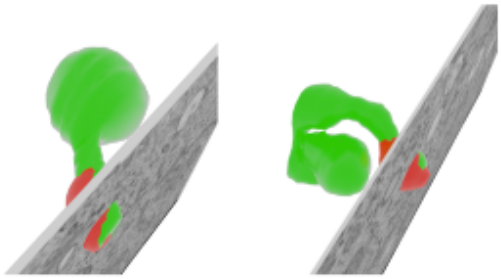


Automatic segmentation detected an early endosome organelle that the manual segmentation ignored



Automatic segmentation detected a lysosome organelle that the manual segmentation ignored

Figure 20: Manual missed classification



automatic: late endosome (red); manual: mitochondria (green); correct: mitochondria

Figure 21: Failed complex objects

singular object, or mislabelled the organelle, as can be seen in Figure 21. The "bridge" parts of the objects proved to be problematic, as they are usually very thin in size and blend in to the background pixels. The workflow was precise enough to detect each individual part of the complex object. For visual clarity, only the first detected part of the complex object is shown. The issue is easily remedied with the eye-test, and manually joining all of the separate parts of the complex object into one whole by adapting the filtering of the morphological image operations (see Methods). However, automating this particular process has proven to be a difficult task, and as such remains unsolved in this version of the workflow. This issue was only relevant in the yeast dataset, and also explains the large number of small objects identifed by FAMOUS but not by the manual workflow. Specifically golgi, multivesicular bodies and liquid droplets had the mentioned issues, as these structures are complex and have many interconnections that FAMOUS did not detect.

The problem with any manual segmentation, is the human factor. vEM image data usually consists of hundreds to thousands of images that need to be analysed. Such work is usually done by students, who may get only a short briefing and whose judgment must be relied upon. Differences in performance are to be expected. Often, not even the evaluation of the structures to be identified is the biggest problem, but the completeness of the evaluation. Many organelles are overlooked. Certainly, the efficiency of manual segmentation also depends on the equipment, a person with a high-quality graphics tablet will get better results than someone with a small screen and a computer mouse. We consider it of outmost importance to hence 'objectify' the process of organelle segmentation for vEM datasets and think that the FAMOUS pipeline is an important step towards a high-throughput quantitative and standardized analysis of vEM datasets.

Methods

FIBSEM sample preparation and data acquisition

Hela cells were grown on a CryoCapsule [11] in DMEM culture medium containing 10% FBS for 3 days, then vitrified by High Pressure Freezing using an HPM Live µ(refs to add once the chapter is out, find the HPF curves to support paper out).The samples were then freeze substituted in Dry acetone plus 1% H2O, 0,05% Uranyl Acetate and 0.1% Glutaraldehyde X hours at -90°C, warmed up to -45°C at +5°C/hour rate, stay at -45°C for 5 hours, rinsed in dry acetone (3x10min) and impregnated in R221 resin (CryoCapCell, France) for 2 hours at 25%, 50%, 75% in acetone. The temperature was raised to -20°C for the last impregnation in 100% R221 (overnight infiltration followed by a second step in 100% for 2 hours prior to UV polymerization). UV polymerization was conducted for 48hours at -20°C, then the temperature was progressively raised to +20°C at a 5°C/hours rate, and UV was continued for 48hours at +20°C. The samples were then evaluated for ultrastructure preservation by transmission electron microscopy prior to analysis by FIB-SEM.

Focused ion beam scanning electron microscopy (FIB-SEM) data was collected using a Crossbeam 540 FIB-SEM with Atlas 5 for 3-dimensional tomography acquisition (Zeiss, Cambridge). Prior to loading into the SEM, the sample was sputter coated with a 10 nm layer of platinum. The cell of interest was relocated by briefly imaging through the platinum coating at an accelerating voltage of 20 kV. On completion of preparation for milling and tracking, images were acquired at 5 nm isotropic resolution throughout the region of interest, using a 10 µs dwell time. During acquisition the SEM was operated at an accelerating voltage of 1.5 kV with 1 nA current. The EsB detector was used with a grid voltage of 1,200 V. Ion beam milling was performed at an accelerating voltage of 30 kV and current of 700 pA. Prior to segmentation, the dataset was cropped, inverted, and registered (using the plugin 'Linear Stack Alignment with SIFT' [20]). The volume of the final dataset was approximately 346.16 µm 3 (1778 images, 10.22 µm x 3.81 µm x 8.89 µm).

Yeast cell sample preparation and data acquisition

Saccharomyces cerevisiae cells were grown in YPD media with 2% glucose to an optical density (OD600) of 0.5. The cells were the filtered using a 0.22um filter [9] and frozen in a Wohlwend Compact 3. The samples underwent freeze substitution in a Leica AFS2 in 2% uranyl acetate in anhydrous acetone for 1h at -90°C, followed by three washes in acetone and stepwise embedding into Lowicryl HM20 resin at -50°C. Finally, they were polymerised using UV light for 5 days whilst allowing the temperature to reach 20°C. Blocks were sectioned using a Reichert Ultracut S to serial 350nm sections onto formvar-coated copper slot grids, stained with 2% uranyl acetate and Reynold's lead citrate. Gold fiducials (15nm) were added onto both surfaces. Tomograms were acquired using an FEI TF30 at 300kV (University of Colorado Boulder) on a Gatan OneView, at a

pixel size of 0.8578nm. Dual-axis tomograms were acquired over a ±60° range at 1.5° increments. Resulting pixel size after reconstruction: 1.7156nm.

Manual segmentation

To evaluate our automated segmentation approach, the same dataset was also manually segmented using Amira 6.0, Thermofisher software [23], using the brush tool and interpolation function of the segmentation editor. Organelles were identified based on their size, shape, and structure, mainly on the X-Y images, all along the Z axis. The orthoslice view was used to correct the Z-positioning of the labeling when necessary. Each segmented organelle was assigned to a morphological group. When the correct assignment was unclear, the orthoslice view was used to help the segmenting scientist. The final volume classes were exported as *.stl files for quantitative comparison with the automatically segmented organelles and further analysis. The entire manual segmentation and visual examination for the FIBSEM dataset alone took about 200 hours for the segmenting scientist.

FAMOUS segmentation pipeline

On a volumetric set of 1800 successive layers of FIB/SEM input images, we used the YOLOMark user interface to define the object classes. We randomly took 20 images from the dataset, and through the YOLOmark user interface, manually and tightly boxed out every compartment in each image according to the class/morphological group we were expecting the compartment to belong to. This preliminary work is the only one required by the end-user and is achieved in about 4 hours for 10 classes.

We used this classification to train YOLOV4 to identify each individual compartment and assign it to a morphological group. This is the 'Instance Segmentation'. Every organelle is classified and boxed out for each single plane of the stack. Given that we know the layer number for any given 2D organelle instance and the distance in nanometers between layers, we can infer the exact 3D location for each organelle location.

In addition, the workflow is fine-tuned to each morphological group to generate a cloud of points outlining the individual compartment based on a conventional image-processing pipeline. On each layer, each identified structure seeks out for the structures located directly above and below itself and looks for correspondences in class. A larger 3D cloud of points outlining the organelle is then repositioned into the original volume, and post-processing is used to smooth the 3D shapes, remove noise, patch holes and re-assemble the cell compartments. This hybrid method uses the YOLO network to classify and box out each compartment, then apply light weight conventional image processing pipeline to accurately segment each compartment class. The expertise of the biologist is used to identify structures in a reasonable time frame, while the image analyst focuses on YOLO training and class segmentation followed by 3D rendering ready for analysis. The processing power required is contained (one

GPU on a workstation is sufficient), and accurate results are generated within a week for one type of dataset with minor input by the end user.

The computer hardware used in the FAMOUS machine learning and image processing pipeline was a regular desktop Windows machine, with 16GB RAM (DDR3, CL16, 2133Hz), Intel i7 7700K with a clock speed of 4.2 GHz, and an NVIDIA Geforce GTX 1060 GPU with 6GB VRAM.

Image Processing

YOLO is an object detection algorithm, meaning that it is able to draw bounding boxes around positive examples of classes of objects it is searching for, but it is not able to isolate the relevant pixels belonging to the object. We solved this problem using basic image processing techniques. A series of morphological operations (erosion, dilation, Gaussian blurring and thresholding) was used to achieve the separation of foreground and background pixels. Each class of organelle had a custom, yet similar (excluding lysosomes) procedure for extracting pixels that belonged to the organelle in each identified region of interest.

Algorithm 1: Early Endosomes segmentation

input : Image $I_{width,height}$, Set of bounding boxes $BB_{x,y,width,height,class=EarlyEndosomes}$

output: Set of points S of all pixels pertaining to Early Endosomes in image $I_{width,height}$

for *each bounding box* $b \in BB$ **do**
- imageROI IR= set I to Region of Interest (ROI) of b;
- IR = reduce to Grayscale(IR);
- IR = Gaussian Blur(IR);
- IR = erode(IR);
- IR = dilate(IR);
- IR = Otsu Threshold(IR);
- C = the largest connected component in IR;
- S = all points belonging to C;

end

We distinguished between early endosomes that are generally light areas against a dark background and late endosomes, mitochondria and nuclei that were the opposite. It was difficult to consistently morphologically isolate the pixels pertaining to lysosomes due to the nearly imperceptible difference between the foreground and background pixels. We therefore assumed that successfully detected lysosome pixels occupied the ellipse that best fit the bounding box of the YOLO detected instance, as seen in the Algorithm 3.

Algorithm 2: Late Endosome, Mitochondria, Nucleus segmentation

input : Image $I_{width,height}$, Set of bounding boxes $BB_{x,y,width,height,class=EarlyEndosomes}$

output: Set of points S of all pixels pertaining to Late Endosomes or Mitochondria, or the Nucleus in image $I_{width,height}$

for *each bounding box* $b \in BB$ **do**

imageROI IR= set I to Region of Interest (ROI) of b;

IR = bitwise Not(IR);

IR = reduce to Grayscale(IR);

IR = Gaussian Blur(IR);

IR = erode(IR);

IR = dilate(IR);

IR = Otsu Threshold(IR);

C = the largest connected component in IR;

S = all points belonging to C;

end

Algorithm 3: Lysosome segmentation

input : Image $I_{width,height}$, Set of bounding boxes $BB_{x,y,width,height,class=EarlyEndosomes}$

output: Set of points S of all pixels pertaining to Lysosomes in image $I_{width,height}$

for *each bounding box* $b \in BB$ **do**

imageROI IR= set I to Region of Interest (ROI) of b;

IR = reduce to Grayscale(IR);

IR = Gaussian Blur(IR);

IR = erode(IR);

IR = dilate(IR);

IR = Otsu Threshold(IR);

C = the largest connected component in IR;

S = all points belonging to C;

end

Organelle Composition from Layers

The above-described methods of extracting salient pixels from bounding boxes is not without fault but does quickly result in usable 2D points that are assembled into point clouds in 3D space. For each of the 1800 FIBSEM input images, for example,we have n sets of 2D points that correspond to pixels of individual organelle instances, as well as the class of each identified organelle. This information effectively gives us the 3D positions of each point of each organelle in the entire sample. Next, we joined the identified organelle slices between layers into individual, coherent 3D organelles. Each bounding box is assigned an ID

number, where bounding boxes of organelles of the same class that meet the necessary criteria to form part of the same organelle are assigned the same ID number. Algorithm 4 describes this procedure.

The resulting sets of 3D points are referred to as point clouds, since we still do not have complete 3D organelles at this point. Techniques for cleaning noise and outliers are used to create the final set of point clouds. Point clouds are transformed into 3D shapes via the meshing procedure described below.

Algorithm 4: Organelle Composition from Layers

input : Images $I_{1..1800}$, Set of bounding boxes $BB_{layer,x,y,width,height,class}$
output : Set of object labels L for all organelle Bounding boxes BB
parameters: integer $layersToScanAboveMe = 50$, double $tolleranceFromCenter = 0.2$

$labelIndex = 0$;
for *each* $image_i \in I_{1..1800}$ **do**
  for *each bounding box* $b \in BB$ *in* $image_i$ **do**
    bounding boxes $bb_{templist}$= get all bounding boxes from BB where
    $BB_{layer} < i$ AND $BB_{layer} > i - layersToScanAboveMe$ AND
    $|((BB_x + BB_{width})/2) - ((b_x + b_{width})/2)| <$ $(tolleranceFromCenter * MAX(BB_{width}, b_{width})/2)$ AND
    $|((BB_y + BB_{height})/2) - ((b_y + b_{height})/2)| <$ $(tolleranceFromCenter * MAX(BB_{height}, b_{height})/2)$;
    if $bb_{templist}$ *Not Empty* **then**
      b_{label}=label of first element of $bb_{templist}$;
    else
      $b_{label} = labelIndex$;
      $labelIndex++$;
    end
  end
end

Cleaning point cloud noise

The output of the network is a set of 3D points, known as a point cloud. Every point is described by 4 parameters: the x,y,z coordinates in 3D space, as well as the normal vector direction of the point. Creating watertight 3D objects from such point clouds requires the use of surface reconstruction algorithms. Such algorithms are extremely sensitive to noise and outliers in the data. Due to this, a pre-processing of the data was implemented before the reconstruction was started. Each point in the point cloud can be described by the number of other points that surround it - neighbouring points. Statistical analysis of the point clouds, per class, output an average distance to neighbouring points.

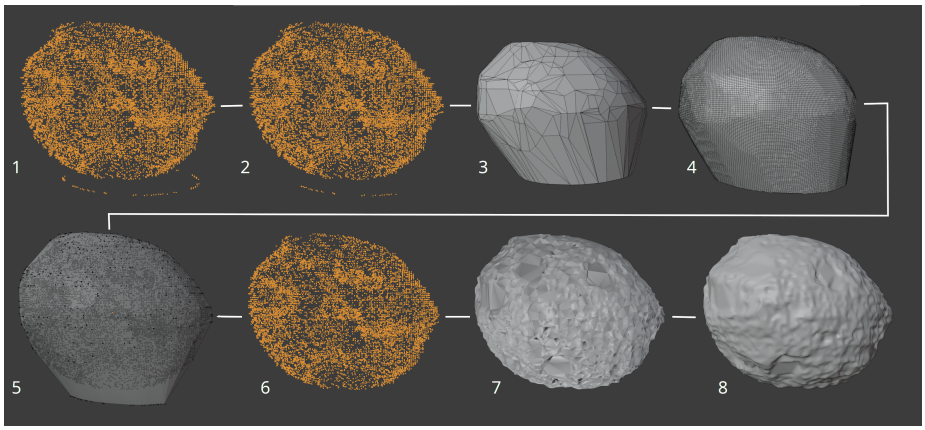


Figure 22: Process of noise removal and reconstruction

Using this number as a threshold, points that do not meet the criterion for the distance are flagged as outliers and removed from the point cloud. This ensures that the sparsest parts of the point cloud are removed and will not influence the reconstruction. This step is done on a per class basis and outputs processed point clouds that can be used for further 3D reconstruction.

Surface reconstruction

The next step in the process consists of generating a single, watertight 3D representation of the point cloud. To generate the 3D representation, the point clouds are imported into Blender [8], a free, open source software for general work with 3D objects. A 3D object can be described as a set of points, edges and faces that define the shape of the object. A singular term for these building blocks of the object is object geometry. The number and distribution of these elements define the complexity and quality of the object itself.

Element number 1 in figure 22 shows an example of raw point cloud data that was generated from the workflow with element number 2 showing the result of the initial, neighbour based, point removal. As the figures show, only the most extreme outliers in the point cloud were identified and removed, leaving noise that was not detected as such still present in the point cloud. Such points did not meet the criterion that was described in the Cleaning point cloud noise subsection. 3D Meshing was achieved through a 3 step process of noise clearing.

The first step in the reconstruction was generating a rough approximation of the point cloud surface as a 3D mesh using Convex Hull operation. The Convex Hull of a set of points P represents the smallest convex set containing P, thus enveloping all of the points of the point cloud with a 3D mesh. Convex Hull trades precision for speed, thus it is prone to creating undesirable 3D artefacts in the reconstructed mesh is shown as element number 3 in figure 22. To resolve this issue, a remeshing algorithm was introduced. The process of remeshing

changes the geometric layout of an object, without changing the shape of the object. Element number 4 in figure 22 shows the differences between the initial Convex Hull geometry and the remeshed geometry. Improved geometry allows for more complex deformations of an object. We used Blender's voxel remesh implementation that uses OpenVDB [17] to generate a new manifold mesh from the input geometry.

In the second step the point cloud and the remeshed Convex Hull were loaded into the same environment and overlayed on top of each other as can be seen in element 5 in figure 22. After which, depending on the object shape, either the rough approximation is scaled by a dynamically calculated amount (1-3% of the full scale), or the rough approximation is projected onto the point cloud before the scaling is done. The object shapes where points are distributed in an uniform manner relative to the center of the object (i.e. all points are at relatively the same distance from the center) use the former, other objects use the latter. Projecting a 3D mesh onto another object is the process where the geometry of the mesh is deformed to the shape of the object on which the projection is being done in a gift-wrapping manner. The point cloud itself will serve as the underlying object around which the 3D mesh will be deformed. The remeshing step is what enables the projection to be successful, as the projection is directly dependent on the geometry layout of the object.

In either case, the rough approximation was scaled and a number of points of the point cloud were exposed. The point cloud is now divided into interior and exterior points in regards to the convex hull approximation. The mesh projection is done once again, ignoring the exterior points thus eliminating any severe noise that remained in the point cloud. A visualization of the resulting point cloud is shown in figure 22 as element number 6.

In the final step the Convex Hull of the cleaned-up point cloud was again calculated. In this part of the pipeline, the projection of the Convex Hull onto the point cloud cannot be omitted. As explained previously, if the projection is to be done, the Convex Hull mesh needs to be remeshed. In this case the remeshing was done to create a more dense geometry i.e. a geometry that can be deformed to a larger extent thus allowing for more detailed surface reconstruction. Once that step was completed, the mesh was projected onto the point cloud, as is shown in figure 22 as element number 7.

As the figure shows, the mesh was deformed to every surface imperfection. However there still existed sharp edges on the mesh, that did not accurately represent the contour of the point cloud locally. We implemented a smoothing algorithm after the projection was completed. The final result of the reconstruction is shown as element number 8 in figure 22.

Acknowledgments

This article is based upon work from COST Action CA17121, supported by COST (European Cooperation in Science and Technology): www.comulis.eu Electron Microscopy STP at the Francis Crick Institute [24]

The work of C.J.P. was supported by the Francis Crick Institute, which receives its core funding from Cancer Research UK (FC001999), the UK Medical Research Council (FC001999), and the Wellcome Trust (FC001999)
This work was supported by a grant from Knut och Alice Wallenbergs Stiftelse (2017.0091) and Swedish Research Council grant 2019-04004 to J.L.H.

References

[1] ariadne.ai - ai-powered biomedical image analysis. `https://ariadne.ai/`, 2022.

[2] Etch a cell. `https://www.zooniverse.org/projects/h-spiers/etch-a-cell`, 2022.

[3] Imaris microscopy image analysis software. `https://imaris.oxinst.com/`, 2022.

[4] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Johannes Schindelin, Albert Cardona, and H. Sebastian Seung. Trainable segmentation: Release v3.1.2. Aug 2016.

[5] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I. Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, September 2019.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[7] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):1–11, 2006.

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021.

[9] Rubai Ding, Kent L McDonald, and J Richard McIntosh. Three-dimensional reconstruction and analysis of mitotic spindles from the yeast, schizosaccharomyces pombe. *The Journal of cell biology*, 120(1):141–151, 1993.

[10] Eva Gagyi, Bernadett Kormos, Karla J. Castellanos, Klara Valyi-Nagy, Dennis Korneff, Patrizia LoPresti, Randy Woltjer, and Tibor Valyi-Nagy.

Decreased oligodendrocyte nuclear diameter in alzheimer's disease and lewy body dementia. *Brain Pathology*, 22(6):803–810, 2012.

[11] Xavier Heiligenstein, Jérôme Heiligenstein, Cédric Delevoye, Ilse Hurbain, Sabine Bardin, Perrine Paul-Gilloteaux, Lucie Sengmanivong, Gilles Régnier, Jean Salamero, Claude Antony, and Graca Raposo. The cryocapsule: Simplifying correlative light to electron microscopy. *Traffic*, 15(6):700–716, 2014.

[12] Larissa Heinrich, Davis Bennett, David Ackerman, Woohyun Park, John Bogovic, Nils Eckstein, Alyson Petruncio, Jody Clements, C. Shan Xu, Jan Funke, Wyatt Korff, Harald F. Hess, Jennifer Lippincott-Schwartz, Stephan Saalfeld, Aubrey V. Weigel, and COSEM Project Team. Automatic whole cell organelle segmentation in volumetric electron microscopy. *bioRxiv*, 2020.

[13] Erlend Hodneland, Tanja Kögel, Dominik Frei, Hans-Hermann Gerdes, and Arvid Lundervold. Cellsegm - a matlab toolbox for high-throughput 3d cell segmentation. *Source code for biology and medicine*, 8:16, 08 2013.

[14] Johanna L. Höög, Cindi Schwartz, Angela T. Noon, Eileen T. O'Toole, David N. Mastronarde, J. Richard McIntosh, and Claude Antony. Organization of interphase microtubules in fission yeast analyzed by electron tomography. *Developmental Cell*, 12(3):349–361, 2007.

[15] James R Kremer, David N Mastronarde, and J Richard McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.

[16] Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Graham Knott, and Pascal Fua. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *IEEE Transactions on Medical Imaging*, 31(2):474–486, 2012.

[17] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. Openvdb: an open-source data structure and toolkit for high-resolution volumes. 07 2013.

[18] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[20] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

[21] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature methods*, 9(7):671–675, 2012.

[22] Mojtaba Seyedhosseini, Mehdi Sajjadi, and Tolga Tasdizen. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2168–2175, 2013.

[23] DETLEV STALLING, MALTE WESTERHOFF, and HANS-CHRISTIAN HEGE. 38 - amira: A highly interactive system for visual data analysis. In Charles D. Hansen and Chris R. Johnson, editors, *Visualization Handbook*, pages 749–767. Butterworth-Heinemann, Burlington, 2005.

[24] Andreas Walter, Gerard J. Kleywegt, and Paul Verkade. Chapter 17 - correlative multimodal imaging: Building a community. In Thomas Müller-Reichert and Paul Verkade, editors, *Correlative Light and Electron Microscopy IV*, volume 162 of *Methods in Cell Biology*, pages 417–430. Academic Press, 2021.

[25] Andreas Walter, Julia G Mannheim, and Carmel J Caruana, editors. *Imaging Modalities for Biological and Preclinical Research: A Compendium, Volume 1*. 2053-2563. IOP Publishing, 2021.

[26] Andreas Walter, Julia G Mannheim, and Carmel J Caruana, editors. *Imaging Modalities for Biological and Preclinical Research: A Compendium, Volume 2*. 2053-2563. IOP Publishing, 2021.

[27] Tao Zeng, Bian Wu, and Shuiwang Ji. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics*, 33(16):2555–2562, 03 2017.

[28] Xiongwei Zhu, George Perry, Mark A Smith, and Xinglong Wang. Abnormal mitochondrial dynamics in the pathogenesis of alzheimer's disease. *Journal of Alzheimer's Disease*, 33(s1):S253–S262, 2013.