Geochemistry, Geophysics, Geosystems

Supporting Information for

U-Net Segmentation Methods for Variable-Contrast XCT Images of Methane-Bearing Sand

F. J. Alvarez-Borges[1], O. N. F. King[1], B. N. Madhusudhan[2], T. Connolley[1], M. Basham[1 3] & S. I. Ahmed[1]

[1]Diamond Light Source Ltd; [2]University of Southampton; [3]The Rosalind Franklin Institute

Contents of this file

Introduction

This supporting information complements descriptions of the tomographic reconstruction, post-processing and U-Net segmentation techniques included in the main text. Thus, the present supporting text, images and tables should be read in conjunction with the main article.

Text S1.

The list of plugins used in the tomographic reconstruction pipeline implemented in Savu 2.4 is here presented (Wadeson et al., 2019; see also Atwood et al., 2015, Wadeson & Basham, 2016). Note that this list has been retrieved from the Savu 2.4 configuration tool.

1. *NxtomoLoader(savu.plugins.loaders.full_field_loaders.nxtomo_loader) – A class for loading standard tomography data in Nexus format.*
2. *DarkFlatFieldCorrection(savu.plugins.corrections.dark_flat_field_correction) – A Plugin to apply a simple dark and flatfield correction to raw timeseries data.*
3. *DezingerSinogram(savu.plugins.filters.dezinger_sinogram) – A plugin working in sinogram space to removes zingers.*
4. *CcpiRingArtefactFilter(savu.plugins.ring_removal.ccpi_ring_artefact_filter) – A plugin to perform ring artefact removal.* [Note: based on algorithm by Titarenko et al. (2010)].
5. *VoCentering(savu.plugins.centering.vo_centering) – A plugin to find the center of rotation per frame.* [Note: Based on algorithm by Vo et al. (2014)].
6. *PaganinFilter(savu.plugins.filters.paganin_filter). A plugin to apply the Paganin filter.* [Note: based on algorithm by Paganin et al. (2002). Used on the phase retrieval pipeline only].
7. *AstraReconGpu(savu.plugins.reconstructions.astra_recons.astra_recon_gpu) – Wrapper around the Astra toolbox for gpu reconstruction.* [Note: Based on algorithms by Palenstijn et al. (2011) and van Aarle et al. (2016); the reconstruction algorithm used was filtered back-projection].
8. *TiffSaver(savu.plugins.savers.tiff_saver) – A class to save output in tiff format.*

Text S2

This is a description of the filtering and volume averaging post-processing procedure, carried out using Fiji (Schindelin et al., 2012; Schneider et al., 2012). It consisted of:

1. The application of a median filter of kernel size 3×3 to the absorption volume and the halving of the resulting greyscale values.
2. The application of unsharp masking of radius 3 and weight 0.70 on the phase contrast volume.
3. The elementwise averaging of both volumes.
4. The application of Fiji auto-contrast enhancement function.

Text S3.

This is a description of the beam hardening reduction procedure. It consisted in the convolution of each reconstructed image slice (horizontal or XY plane) with a mollifier function of the same size and approximately the inverse shape to that of the 'cupping' artefact, as shown in Figure S1. The mollifier function takes the radius of the field of view (r) and two fitting constants ($a = 0.10$ and $b = 0.52$) as parameters. The a and b parameters were arbitrarily adjusted until achieving a shape that matched the cupping effect (Figure S1c). A second mollifier function using $a = 0.023$ and $b = 32$ was convolved solely with the non-sand phase, retrieved via simple thresholding, as this phase exhibited a stronger cupping effect. A circular mask with a radius of 1100 pixels was then applied

to each slice to remove voxels at the edges of the FOV which were resistant to cupping correction. The mollifier function has the form:

$$f(x,y) = \exp\left[-a/(b - r^2) - (a/b)\right] \quad \textbf{(1)}$$

and was plotted across a 2560×2560-pixel grid for convolution with the tomographic slices, as shown in Figure S1b. The procedure was implemented using MATLAB code available from King & Alvarez-Borges (2021).

Text S4.

This is a description of the volume averaging procedure used for the 2D multilabel and multi-plane segmentation strategy. The 1554×1554×2000 SRXCT data was sliced in the XY, XZ and YZ planes and the segmentation predicted for all image slices, as presented in the main text. Then, the data volume was rotated 90 degrees around the 4-fold symmetry axis running perpendicular to the XY plane. The plane-wise slicing, segmentation and rotation was then repeated until the original data volume had been rotated 270 degrees and 12 separate volumes of labelled data had been predicted. Each of the predicted volumes was then split into two separate binary predictions, one containing labels for sand (label value of 1) vs background (label value of 0) and the other containing labels for CH_4 gas (label value of 1) vs background (label value of 0). These respective binary segmentations were then summed, giving two label volumes with voxel values in the range 0 to 12, one volume for CH_4 vs background and the other for sand vs background. A manual threshold was then applied to these summed volumes such that there was an agreement between y or more predictions for the sand vs background volume, and between 11 or more predictions for the CH_4 vs background volume.

Text S5.

This is a description of the procedure used to create arbitrarily sparsely annotated data from densely annotated images, to be used for RootPainter training. For each 2D label image:

1. A copy was created in which all the label values were set to 0 except for the CH_4 phase (or the label of interest), from which the foreground annotations will be created.
2. CH_4 label pixels were morphologically dilated by 100 pixels. The pixels created by the dilation procedure were then re-labelled as background while the original CH_4 label pixels were kept as foreground. This effectively created RootPainter background regions around the CH_4 labels. These background regions included both brine and sand areas from the tomography slices.
3. The rest of the image was set as RootPainter 'empty annotation'.

This method was implemented in MATLAB® and is available from King & Alvarez-Borges (2021).

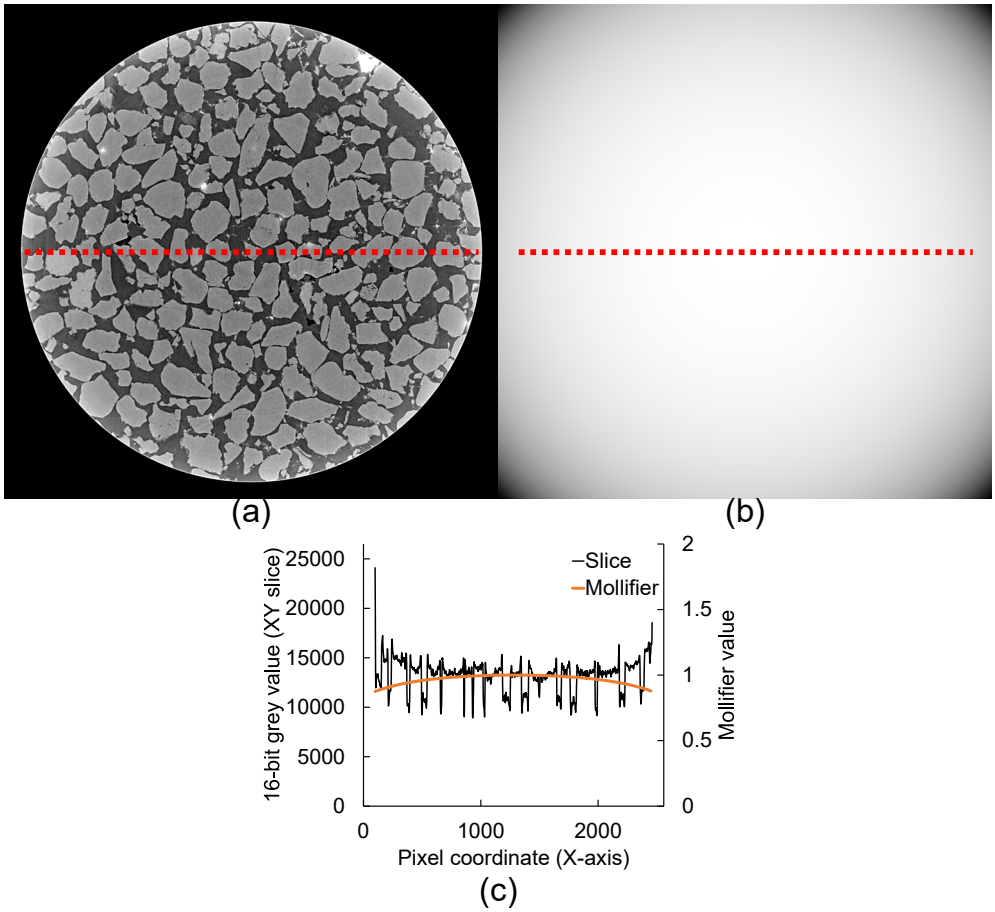

Figure S1. (a) Central slice of dataset 89075 after the first post-processing stage (Fiji-based corrections); **(b)** 2D plot of mollifier function for beam hardening artefact reduction; **(c)** Horizontal pixel value (X-axis) profiles of slice and mollifier function (profile sampling location indicated by red dotted lines).

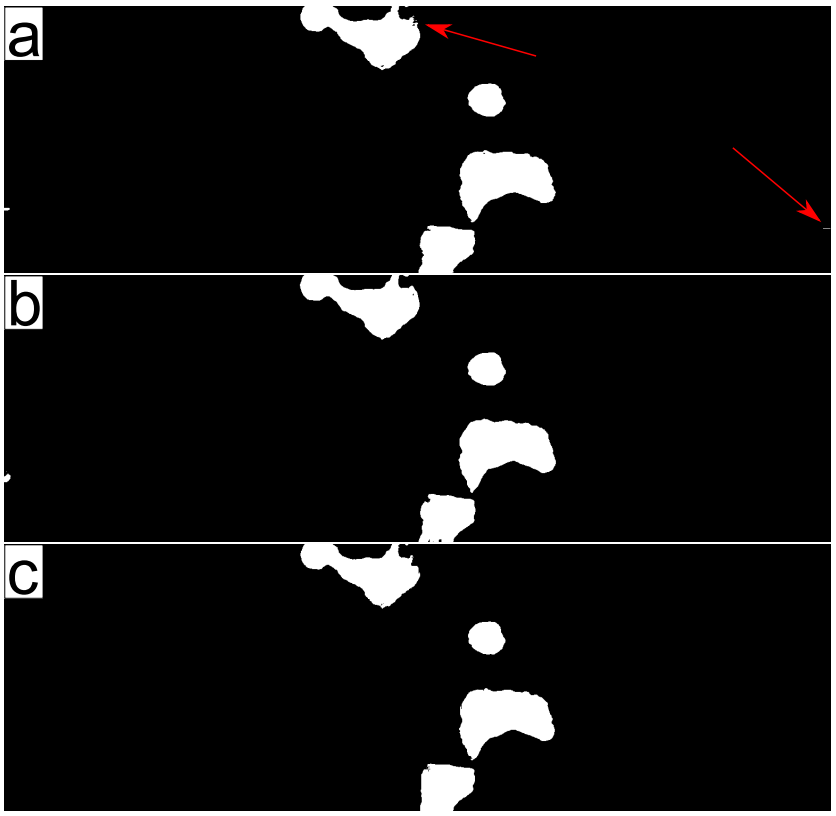

Figure S2. YZ (vertical) section of the central 500 XY slices of the segmented 89062 data set (CH_4 gas label only) using **(a)** RootPainter, **(b)** the 3D hierarchical method and **(c)** the 2D multilabel multiplane method. Horizontal stripe artefacts indicated with arrows.

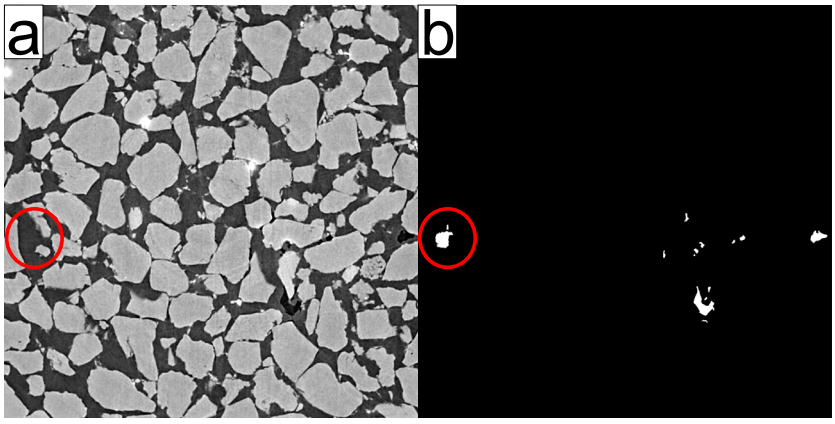

Figure S3. 89090 **(a)** original and **(b)** segmented slice showing the lowest performance metrics of Figure 7(f) of the main text (corresponding to RootPainter) due to a cluster of false positive pixels (marked by circle.

Training data source	Labels	Number of training epochs	Final training loss (BCE)	Final validation loss (BCE)	Final validation metric (IOU)
89062 (384x384x384)	CH_4 vs Background	94	0.0313	0.0237	0.935
	Sand vs Background	83	0.0317	0.0332	0.977
89062 (572x572x572)	CH_4 vs Background	84	0.00551	0.0307	0.918
	Sand vs Background	85	0.0426	0.0290	0.980
89069 (384x384x384)	CH_4 vs Background	82	0.0178	0.0371	0.759
	Sand vs Background	69	0.0305	0.0471	0.957
Notes: BCE – Binary Cross Entropy; IOU – Mean Intersection Over Union.					

Table S1. Binary 3D U-Net training metrics.

Training data source	Approach	Number of training epochs	Final training loss (CE)	Final validation loss (CE)	Final validation metric (%)
89062 (572x572x572)	Single-plane	15	0.020	0.014	99.48
	Multi-plane	15	0.019	0.012	99.59
Notes: CE – Cross Entropy; % – Percentage of correctly labelled voxels.					

Table S2. Single- and multi-plane 2D multilabel U-Net training metrics.

References

Atwood R.C., Bodey A.J., Price S.W.T., Basham M. & Drakopoulos M. (2015) A high-throughput system for high-quality tomographic reconstruction of large datasets at Diamond Light Source. *Philosophical Transactions of the Royal Society A, 373 (2043)*, 2369–2393. https://doi.org/10.1098/rsta.2014.0398

King O.N.F. & Alvarez-Borges F.J. (2021) Gas Hydrate Segmentation Using U-Nets. Code Repository. https://doi.org/10.5281/zenodo.4683002

Paganin D., Mayo S.C., Gureyev T.E., Miller P.R. & Wilkins S.W. (2002) Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous

object. *Journal of Microscopy, 206 (1)*, 33-40. https://doi.org/10.1046/j.1365-2818.2002.01010.x

Schindelin J., Arganda-Carreras I., Frise E., Kaynig V., Longair M., Pietzsch T., et al. (2012) Fiji: an open-source platform for biological-image analysis. *Nature Methods, 9 (7)*, 676-82. https://doi.org/10.1038/nmeth.2019

Schneider C.A., Rasband W.S. & Eliceiri K.W. (2012) NIH Image to ImageJ: 25 years of image analysis. *Nature Methods, 9 (7)*, 671-675. https://doi.org/10.1038/nmeth.2089

Titarenko V., Bradley R., Martin C., Withers P. & Titarenko S. (2010) Regularization methods for inverse problems in x-ray tomography. *SPIE Optical Engineering + Applications* (7804). Society of Photo-Optical Instrumentation Engineers (SPIE). https://doi.org/10.1117/12.860260

Vo N.T., Drakopoulos M., Atwood R.C. & Reinhard C. (2014) Reliable method for calculating the center of rotation in parallel-beam tomography. *Optics Express, 22 (16)*, 19078-19086. https://doi.org/10.1364/OE.22.019078

Wadeson N. & Basham M. (2016) Savu: A Python-based, MPI Framework for Simultaneous Processing of Multiple, N-dimensional, Large Tomography Datasets. arXiv:1610.08015 [Preprint]

Wadeson N., Basham M., Parsons A., Kazantsev D., Vo N.T., Schoonjans T., et al. (2019) DiamondLightSource/Savu: Version 2.4. https://doi.org/10.5281/zenodo.3541873