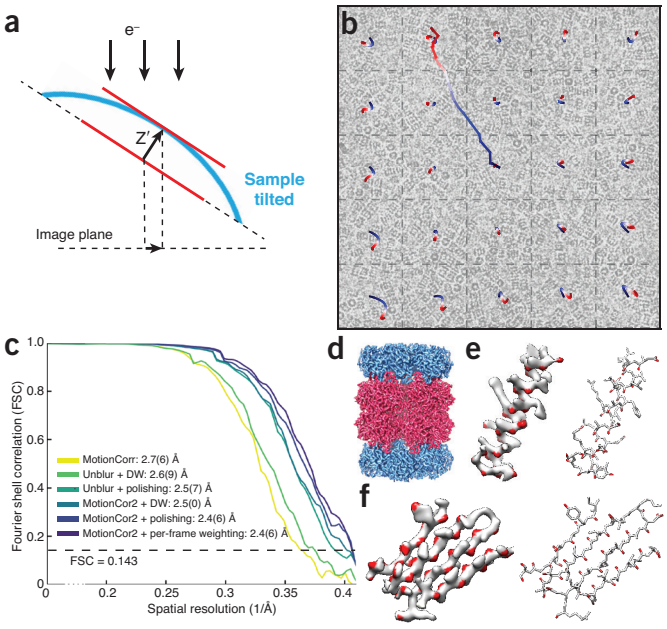

Figure 1 | Local motion correction by MotionCor2. (**a**) Schematic drawing illustrates that when the sample is tilted the observable motion in the image plan is the projection of *z*-motion produced by doming of the sample under electron beam. (**b**) Image of frozen hydrated archaeal 20S proteasome overlaid with the traces of global motion based upon whole-frame alignment (long trace originated from the center of image) and each patch predicted from the polynomial function. (**c**) Fourier shell correlation (FSC) curves of 3D reconstructions determined using micrographs corrected by Unblur with dose weighting, Unblur followed by particle polishing, correction by MotionCor2 with dose weighting, MotionCor2 followed by particle polishing and MotionCor2 with per-frame B-factor weighting. (**d**) 3D reconstruction of archaeal 20S proteasome filtered to 2.5-Å resolution and sharpened by a temperature factor of −103.8 Å^2. (**e**) Density of an α helix from the map, with resolved oxygen atom functional groups colored in red. Visualization of main chain carbonyls requires resolution below 3 Å. The refined atomic model is shown side by side for comparison. (**f**) As in **e**, but showing a β sheet.

computationally intensive particle polishing in RELION can be skipped. Importantly, it also works on a wide range of data sets, including cryo tomographic tilt series.

Data availability statement. The refined coordinates of archaeal 20S proteasome and the density maps of both archaeal 20S proteasome and TRPV1 are included as **Supplementary Data**.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper (doi:10.1038/nmeth.4193).

ACKNOWLEDGMENTS

We thank X. Li for helpful discussion during the initial stage of this work. We also thank M. Braunfeld for supporting the cryo-EM facility at UCSF, G. Greenan (Department of Biochemistry and Biophysics, University of California San Francisco) for providing his cryo-tomographic tilt series collected on a *Drosophila* centriole, and C. Kennedy for supporting the computational infrastructure for processing cryo-EM data. This work was supported in part by grants from National Institute of Health—R01GM031627 to D.A.A. and P01GM111126, P50GM082250, R01GM082893 and R01GM098672 to Y.C. Y.C. and D.A.A. are Investigators of Howard Hughes Medical Institute.

AUTHOR CONTRIBUTIONS

S.Q.Z. implemented the algorithm and wrote all codes for MotionCor2. D.A.A. contributed to algorithm development. S.Q.Z, E.P., Y.C. and D.A.A. designed experiments to evaluate the performance of MotionCor2. K.A.V. designed initial experiments for camera defect correction. S.Q.Z. and E.P. carried out image processing. E.P. and J.-P.A. collected low-defocus cryo-EM images of 20S proteasomes. S.Q.Z., E.P., Y.C. and D.A.A. wrote the manuscript. All authors participated in discussion and revision of the manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Shawn Q Zheng[1,2], Eugene Palovcak[1], Jean-Paul Armache[1], Kliment A Verba[1], Yifan Cheng[1,2] & David A Agard[1,2]

[1]Department of Biochemistry and Biophysics, University of California, San Francisco, San Francisco, California, USA. [2]Howard Hughes Medical Institute, University of California, San Francisco, San Francisco, California, USA.
e-mail: ycheng@ucsf.edu or agard@msg.ucsf.edu.

Published online 27 February 2017; doi:10.1038/nmeth.4193

1. Cheng, Y. *Cell* **161**, 450–457 (2015).
2. Kühlbrandt, W. *Science* **343**, 1443–1444 (2014).
3. Brilot, A.F. *et al. J. Struct. Biol.* **177**, 630–637 (2012).
4. Li, X. *et al. Nat. Methods* **10**, 584–590 (2013).
5. Liao, M., Cao, E., Julius, D. & Cheng, Y. *Nature* **504**, 107–112 (2013).
6. Bai, X.C., Fernandez, I.S., McMullan, G. & Scheres, S.H. *eLife* **2**, e00461 (2013).
7. Rubinstein, J.L. & Brubaker, M.A. *J. Struct. Biol.* **192**, 188–195 (2015).
8. Grant, T. & Grigorieff, N. *eLife* **4**, e06980 (2015).

Automatic tracing of ultra-volumes of neuronal images

To the Editor: Despite substantial advancement in the automatic tracing of neuronal morphology in recent years[1–3], it is challenging to apply the existing algorithms to large image data sets containing billions or even trillions of voxels. Most neuron-tracing methods published to date were not designed to handle such data. We introduce UltraTracer (**Fig. 1**), a solution designed to extend any base neuron-tracing algorithm to allow the tracing of ever-growing data volumes. We applied this approach to neuron-tracing algorithms with different design principles and tested it on human and mouse neuron data sets that have hundreds of billions of voxels. Results indicate that UltraTracer is scalable, accurate, and more efficient than other state-of-the-art approaches.

The core algorithm of UltraTracer (**Fig. 1**) reconstructs a neuron structure from the available image data on the basis of a formulation of maximum-likelihood estimation. The underlying assumption is that the occurrence of a specific neuron structure could be modeled using the joint probability of all of its subparts given the image. Briefly, UltraTracer iteratively factorizes the joint probability based on progressive maximization of conditional probabilities of the occurrence of salient and continuous subparts of a neuron (**Supplementary Note**). Therefore, UltraTracer explores an image by following where the neurite signal goes, on the basis of either adaptive windows generated based on the already reconstructed neuron structure or certain domain knowledge (prior information or statistics) of neuron morphology, to help refine the choice of the next tracing subarea (**Supplementary Note**). This process repeats until the neuron structure grows as completely as possible. We designed the UltraTracer software to quickly extract an arbitrary subvolume of interest from large neuron image files (**Supplementary Note**), and thus smoothly traced an image archive without the need to load a large number of image voxels into computer memory.

As a crucial utility that was not previously available to reconstruct large-scale data sets, UltraTracer extends an arbitrary base tracer to make it possible to trace an ever-increasing image volume. We tested this by considering ten representative base tracing algorithms ported to BigNeuron[3] (https://github.com/BigNeuron/BigNeuron-Wiki/wiki/Neuron-Reconstruction-Algorithms) that have different design principles, performances, and output formats (**Supplementary Figs. 1**, **2**, and **3**; **Supplementary Note**). The performance gain of UltraTracer over the direct use of certain base tracers was within the range of 3–6 times (**Supplementary Fig. 1b**). UltraTracer results were accurate, as their average spatial distances to independent manual reconstructions were around 3 voxels, comparable to the spatial distance of the manual reconstructions themselves (3.56 voxels) (**Supplementary Fig. 1b**). In addition, for two base tracers, NeuTube[4] and MOST[5], UltraTracer had a gain of 10–30-fold in tracing accuracy (**Supplementary Fig. 1b**). Testing of six other base tracers (**Supplementary Fig. 2**) indicated similar improvement. When a computer with smaller memory was used or the image volume increased greatly, UltraTracer was consistently superior to the conventional approach (**Supplementary Fig. 3**).

The APP2 algorithm[6] was a good base tracer, in terms of speed–accuracy trade-off (**Supplementary Fig. 1**), for both laser-scanning and brightfield images (**Supplementary Figs. 3**–**7**). The APP2-based UltraTracer scaled robustly in tracing the sparse neuronal structures in images with 521 billion voxels, reducing the data volume in tracing between 3 and 40 times (**Supplementary Fig. 3**). Typically a bigger data-volume reduction rate was achieved for a larger image volume. Measured in terms of spatial distance, bifurcation points, and five other morphological and topological features, and compared against the statistics drawn from collections of reconstructions produced using control images (**Supplementary Note**), the accuracy of reconstructions produced by UltraTracer was consistent with that of reconstructions generated using the traditional approach when the image data set could be accommodated by the computer memory (**Supplementary Fig. 3**, bottom left).

We used UltraTracer to combine multiple different base tracers (**Supplementary Note**; **Supplementary Figs. 8** and **9**), for example using APP2 in the soma area while using NeuTube and MOST to trace curvilinear structures. In a more complicated case, for every adaptively searched image region, we profiled the reconstructions generated by several base tracers, and then chose either the best reconstruction or their consensus as the result from the current image region (**Supplementary Fig. 9**). In this way UltraTracer could provide more consistent reconstructions compared to manual work. We also used UltraTracer to reconstruct human neurons, including their axons and dendrites, from separate but serial slices of brain tissue (**Supplementary Note**; **Supplementary Fig. 10**). Additional information about the algorithm can be seen in **Supplementary Figures 11**–**13**.

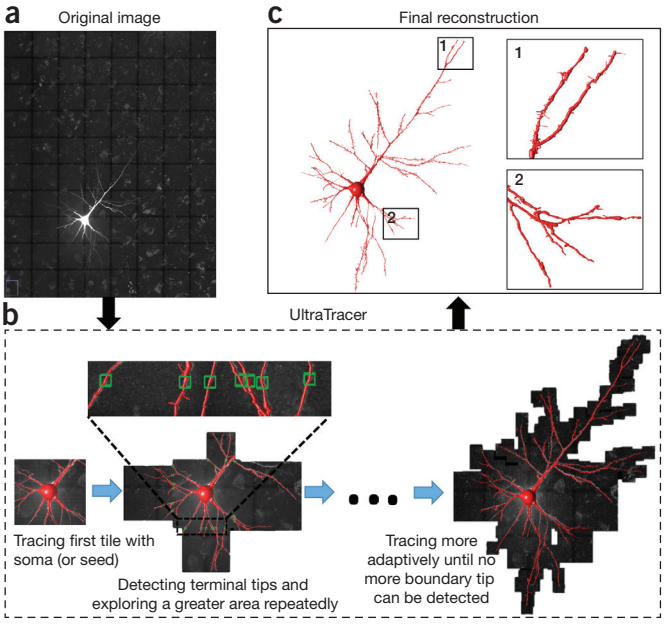

Figure 1 | Workflow of UltraTracer for tracing a large 3D image volume. (**a**) 3D confocal image stack of a Lucifer-Yellow-labeled human pyramidal neuron. The voxel size is 0.18 × 0.18 × 0.5 μm, and the overlaid grid (black lines) indicates how the image volume is subdivided into uniform tiles. (**b**) UltraTracer first traces the subarea containing the soma and then detects the neuron terminal tips in the reconstruction, and adaptively explores and traces neighboring subareas. Green boxes indicate terminal tips detected in tracing a subarea. (**c**) Final reconstruction produced by UltraTracer, with zooms of two parts for detailed visualization.

Data availability. UltraTracer is open source and available in Vaa3D software (vaa3d.org) and as **Supplementary Software**. The sample data are publicly available and can be downloaded from GitHub (https://github.com/Vaa3D/Vaa3D_Data/releases/download/v0.9/ultratracer_testing_data.zip).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank the Allen Institute for Brain Science and data contributors to the BigNeuron project for providing neuron data sets. This work was funded by the Allen Institute for Brain Science. The authors wish to thank the Allen Institute founders, P.G. Allen and J. Allen, for their vision, encouragement and support.

AUTHOR CONTRIBUTIONS

H.P. conceived this project, designed and managed this study, proposed the theoretical framework of the method, and wrote the paper with assistance from Z.Z. and other coauthors. Z.Z. developed the tip-queue-based neuron growth algorithm, implemented the software, and generated results, with the help of coauthors.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Hanchuan Peng[1,5], Zhi Zhou[1,5], Erik Meijering[2], Ting Zhao[3], Giorgio A Ascoli[4] & Michael Hawrylycz[1]

[1]Allen Institute for Brain Science, Seattle, Washington, USA. [2]Erasmus University Medical Center Rotterdam, Rotterdam, the Netherlands. [3]Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, USA. [4]Krasnow Institute for Advanced Study, George Mason University, Fairfax, Virginia, USA. [5]These authors contributed equally to this work.
e-mail: hanchuanp@alleninstitute.org

1. Helmstaedter, M. *Nat. Methods* **10**, 501–507 (2013).
2. Acciai, L., Soda, P. & Iannello, G. *Neuroinformatics* **4**, 353–367 (2016).
3. Peng, H. *et al. Neuron* **87**, 252–256 (2015).
4. Zhao, T. *et al. Neuroinformatics* **9**, 247–261 (2011).
5. Wu, J. *et al. Neuroimage* **87**, 199–208 (2014).
6. Xiao, H. & Peng, H. *Bioinformatics* **29**, 1448–1454 (2013).