Supplementary Information

Integrating Light-Sheet Imaging with Virtual-Reality to Recapitulate Developmental Cardiac Mechanics

Yichen Ding, Arash Abiri, Parinaz Abiri, Shuoran Li, Chih-Chiang Chang, Kyung In Baek, Jeffrey J. Hsu, Elias Sideris, Yilei Li, Juhyun Lee, Tatiana Segura, Thao P. Nguyen, Alexander Bui, René R. Sevag Packard, Peng Fei and Tzung K. Hsiai*

This PDF file includes:

Other Online Supporting Information for this manuscript includes the following:

Figure S1: Comparison between VR-LSFM data and histology

To validate the structures visualized by our VR-LSFM hybrid platform, we performed comparisons (**Figure S1**) of LSFM imaging raw data (left panels A, D and G), LSFM imaging segmented results (middle panels: B, E and H), and histological reference images (right panels: C, F and I). We demonstrate correlations between the segmented GFP-labeled ROMK channels (**Figure S1B**) with GFP-immunostained ROMK channels in histological sections (**Figure S1C**). The histology was obtained from the paraffin sections at 5 μm in thickness from a 16-month-old mouse heart following intravenously injected AAV9-ROMK-GFP to the tail at 4 months (**Figure S1C**). The exogenous fusion ROMK-GFP construct was expressed as a transmembrane ion channel tetramer identified by the brown immunostaining (arrows) in the ventricle. In addition, we validated the anatomical structure of an adult zebrafish heart at 60 days post fertilization (dpf) with the histological result (**Figure S1 D-F**), and an embryonic zebrafish heart at 5 dpf with the H&E staining image (**Figure S1 G-I**). Thus, the VR-LSFM-acquired imaging data demonstrate high spatial resolution as compared to the histological data.

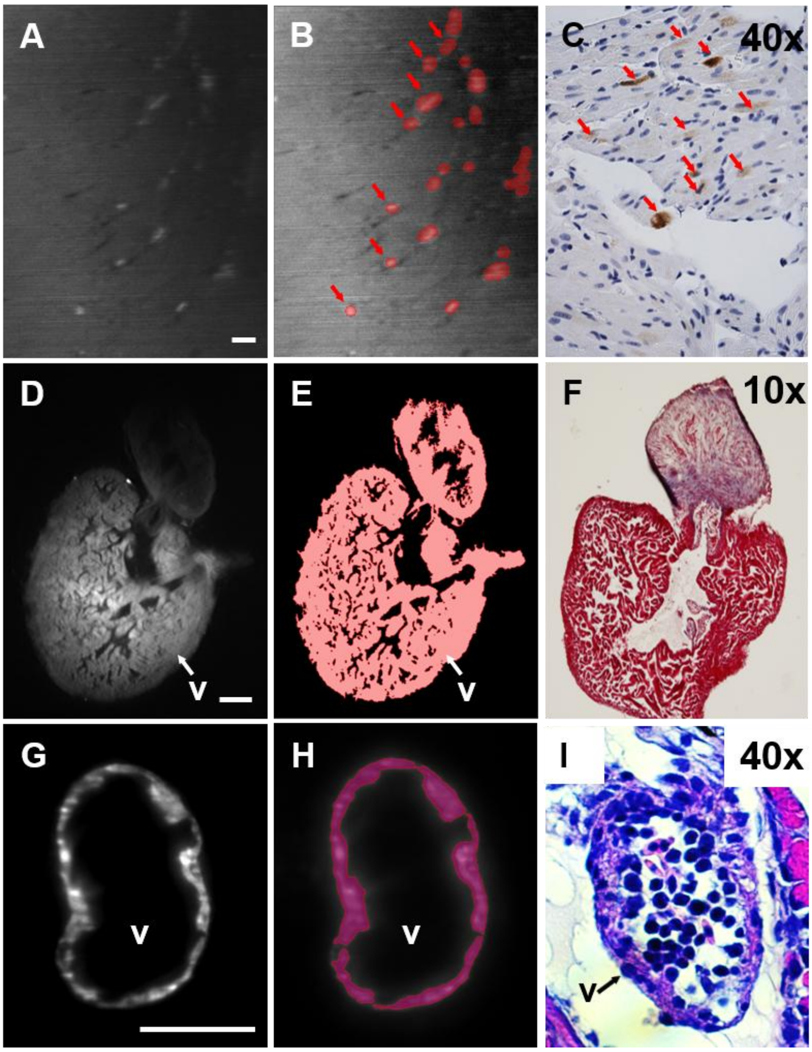

Figure S1. Comparison between VR-LSFM data and histology. VR-LSFM imaging raw data (left panels: A, D and G), segmented results (middle panels: B, E and H), and histological images (right panel: C, F and I). (A-C) Mouse heart; (D-F) adult zebrafish heart at 60 dpf; (G-I) embryonic zebrafish heart at 5 dpf. VR: virtual reality. LSFM: light-sheet fluorescence microscopy. A: atrium. V: ventricle. dpf: days post fertilization. Scale bars: 100 μm. Magnification: (C) 40x. (F) 10x. (I) 40x.

Figure S2: Simplified CLARITY method

The adult hearts were rinsed in phosphate-buffered saline (PBS) three times for 10 minutes and fixed in 4% paraformaldehyde (Electron Microscopy Sciences) at 4°C overnight. Next, the samples were placed in a 4% acrylamide solution along with

0.5% w/v of the photoinitiator 2,2’-Azobis[2-(2-imidazolin-2-yl)propane] dihydrochloride (VA-044, Wako Chemicals USA), followed by overnight incubation at 4°C and incubation at 37°C for 2-3 hours to initiate polymerization of the acrylamide. Once polymerized, the tissues were rinsed with PBS and placed into a clearing solution comprised of 8% w/v sodium dodecyl sulfate (SDS, Sigma Aldrich, USA) and 1.25% w/v boric acid (Fischer, USA) (pH=8.5). The samples were incubated at 37°C until they were clear, followed by one day in 1x PBS to remove residual SDS. The hearts were incubated in refractive-index matching solution (RIMS) until transparent (**Figure S2**). Additionally, RIMS was made of forty grams of Sigma D2158 (Histodenz) in 30 ml of 0.02 M PB with 0.1% tween-20 and 0.01% sodium azide, pH to 7.5 with NaOH.

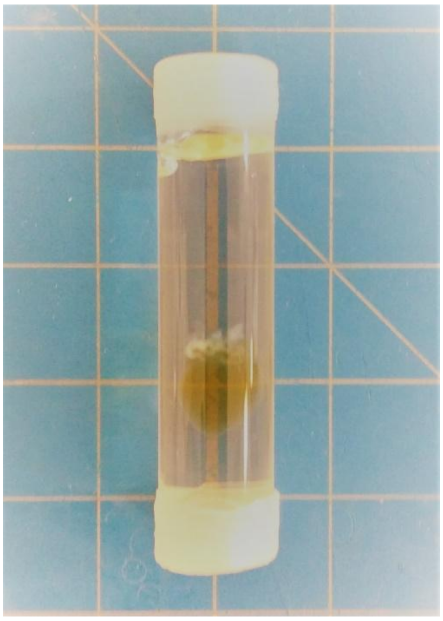

Figure S2. The translucent heart after simplified CLARITY.

Table S1

Procedurals of simplified CLARITY for mouse hearts

Procedural	Time	Note
Rinsed in PBS	~ 10 minutes	3 times
Fixed in 4% paraformaldehyde	Overnight	4° C
Polymerized in 4% acrylamide solution + 0.5% w/v of photoinitiator	Overnight	4° C
	2~3 hours	37° C
Placed in clearing solution	1~3 weeks	37° C
Incubated in RIMS	1~2 days	37° C

Figure S3: Combination of the smartphone and the Google VR viewer

We provided a time chart for representative proceduals (**Figure S3a)**. In the following parts, we took the Google Cardboard and an adult mouse heart as examples to introduce how this VR-LSFM platform was developed. We could communicate with the virtual object by clicking the button on the Cardboard (**Figure S3b**) to launch specific interactions, such as random walking and text reading. To build a VR application on the smartphone, we took the following steps:

a) Generating data. All imaging data used here was from the real biological sample.

a.1) Using LSFM to generate series of 2-D slices.

a.2) Scanning the whole sample step by step, where the step size is entirely dependent on the sample and Nyquist sampling theorem, usually 0.5μm to 5μm. At each step, we could acquire a single 2-D slice, which is *.tif/bmp/jpg/png* image, or any other bitmap file.

a.3) Using different laser lines, we could acquire specific fluorescence and anatomical information.

a.4) After imaging, we exported all the data to our workstation for the rest of the imaging processing. The key parameters in our workstation include the following:

Category	Specifications
CPU	2x Intel Xeon E5 v3, 3.00GHz
RAM	256GB, 931MHz
Graphics	2x NVIDIA Quadro K6000
	2x DELL U2713HM
Storage	24 TB
Motherboard	Dell Inc. 0215PR
System	Windows 8.1 Pro 64-bit

b) Image post-processing. The processing methods can be accomplished in MATLAB, ImageJ and Fiji.

b.1) De-noising the raw data with median filter, mean filter or any other filter which is proper for de-noising the raw data, if necessary.

b.2) Removing stripes from the raw data with variable stationary noise remover (VSNR).

b.3) Applying Gaussian blur method to normalize the distribution of brightness and contrast.

b.4) The output from VSNR was mathematically divided by the result of Gaussian blur method.

b.5) Applying deconvolution approach for the mapping of acoustic sources 3D (DAMAS3) as a deconvolution method with point spread function acquired from the same imaging system.

b.6) Applying the method termed batch intensity normalized segmentation (BINS) to segment the region of interest.

c) Conversion from bitmap images to 3-D editable file. We finished this step in Amira. Nevertheless, it is able to be performed in ImageJ, Fiji and ChimeraX.

c.1) Using processed slices, we reconstructed a 3-D object with proper voxel information, computed the surface, and generated polygon mesh for objects c.2) Extracting the surface and exporting it as an *obj* file.

d) Refinement of polygon mesh. We validated the reduction of data complexity in Amira and edited the mesh in Maya with the license of educational purpose.

d.1) Smoothing the mesh and controling over the number of vertices, edges and faces in Amira, and comparing the polygon mesh with raw data to ensure the applicability of the model for the authentic output.

d.2) Importing the *obj* file into Maya (Autodesk Inc) or 3D Max (Autodesk Inc) for editing and rendering.

d.3) Modifying the scale, transforming the shape, and do more lighting, shading, texturing and rendering work, if necessary.

d.4) Adding scenes and motions in Maya, if necessary.

d.5) Exporting the whole project, including the objects, scenes and rendering information as an *fbx* file to Unity software. In Maya 2016, we could directly send the whole project to Unity (Unity Technologies).

e) Building VR application.

e.1) Downloading Google cardboard second development kit (SDK), including cardboard SDK for Unity, Android SDK and iOS SDK. Installing the application referring to the instruction online:

https://developers.google.com/cardboard/overview.

e.2) Launching a new project, and opening the *fbx* file in Unity, coming with lighting, shading and materials information.

e.3) Importing cardboard SDK for Unity, and replacing the default main camera with the CardboarMain Camera (Assets\Cardboard\Prefabs\CardboardMain). Thus, all the objects in the scene could be displayed in two cameras (left and right).

e.4) Designing the movement (translation-rotation-inversion) of objects, camera and any other related items in the scene.

e.5) Creating the interactive actions with the objects.

e.6) The unmet needs in the default SDK could be achieved by coding custom's own scripts based on C# or Java scripts.

e.7) Choosing "File\Build settings\".

e.8) Choosing platform as "Android" or "iOS".

e.9) Default orientation should be "Landscape left" instead of "Auto rotation".

e.10) Built and ran the new application (**Figure S3b**).

Note:

i.) In the previous version of Maya, we might save the project as *ma* (Maya ASCII) or *mb* (Maya Binary) or *fbx* file as well, and then move all files to the assets of Unity.

ii.) Unity is not the only platform which provides VR development. Any platform which could import data from Maya could be involved, such as Unreal Engine. Unity can be downloaded free of charge for academic use. (https://store.unity.com/education).

iii.) Maya is not the only platform which could provide editing or rendering for the imported file. Any platform which could provide polygon mesh for editing could be involved, such as 3d Max (Autodesk Inc). Maya and 3D Max can be downloaded free of charge for academic use. (https://www.autodesk.com/education/about-autodesk-education).

iv.) Besides Android and iOS, it provides several platforms for final application in Unity, such as Tizen, Xbox 360, Xbox One, PS3, PS4, tvOS, Samsung TV, etc. Any of them could be used as the final platform as long as the hardware is compatible.

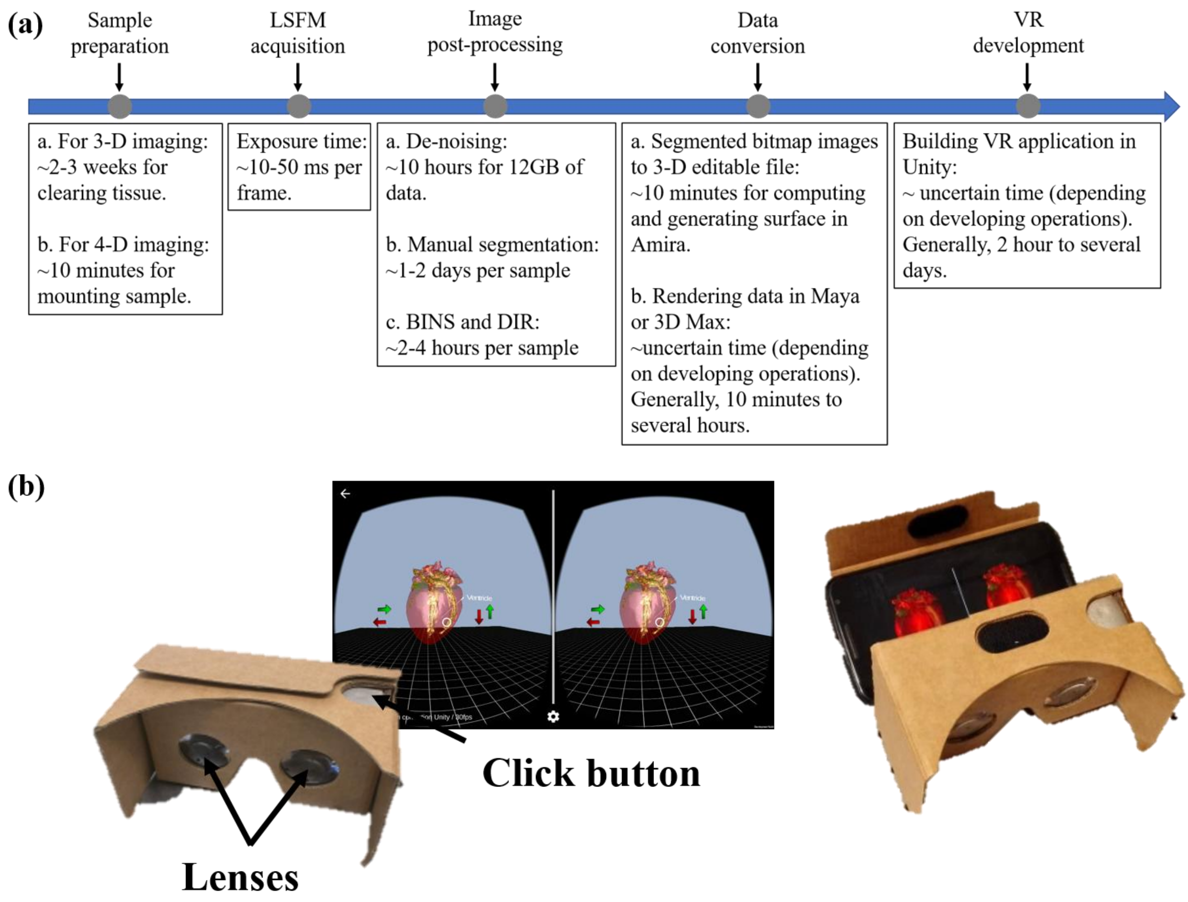
(a)
Sample preparation
LSFM acquisition
Image post-processing
Data conversion
VR development
a. For 3-D imaging: ~2-3 weeks for clearing tissue.
b. For 4-D imaging: ~10 minutes for mounting sample.
Exposure time: ~10-50 ms per frame.
a. De-noising: ~10 hours for 12GB of data.
b. Manual segmentation: ~1-2 days per sample
c. BINS and DIR: ~2-4 hours per sample
a. Segmented bitmap images to 3-D editable file: ~10 minutes for computing and generating surface in Amira.
b. Rendering data in Maya or 3D Max: ~uncertain time (depending on developing operations). Generally, 10 minutes to several hours.
Building VR application in Unity: ~ uncertain time (depending on developing operations). Generally, 2 hour to several days.
(b)
Click button
Lenses

Figure S3. Detailed procedurals and processing time for building a VR-LSFM application. (a) A time chart of sequential proceduals from sample preparation to VR development. (b) Google Cardboard and the VR-mode display on the smartphone. VR: virtual reality. LSFM: light-sheet fluorescence microscopy.

Figure S4: Light-sheet fluorescence imaging system

The light-sheet fluorescence microscopy (LSFM) system was built to generate a dual-sided illumination and dual-channel detection units as follows (**Figure S4**):

a) Illumination unit. All of the collimated lasers at 405 nm, 473 nm, 532 nm and 589 nm were precisely aligned to a single beam after passing through a set of beam splitters (BS1, BS2 and BS3) and a mirror (M1). The Gaussian laser beams were expanded (BE), splitted (BS4), and guided towards two identical illumination arms. This beam was then incident up on the achromatic cylindrical lens (CL I or II) and relay lens (I or II). Finally, the beam was guided toward the illumination lens to generate a thin light-sheet. The cylindrical and illumination lenses were placed on a piezoelectric actuator that can be mobilized along the beam propagation axis to provide an optimal line of focus.

ii) Detection unit. There were two sCMOS cameras, a set of filters (FW I and II), a dichroic mirror (DC), a tube lens (TL), and an objective lens (DL at low magnification: 4X/0.13 or 10X/0.3). A five-axis translational stage will be customized to set the scanning direction and sample orientation.

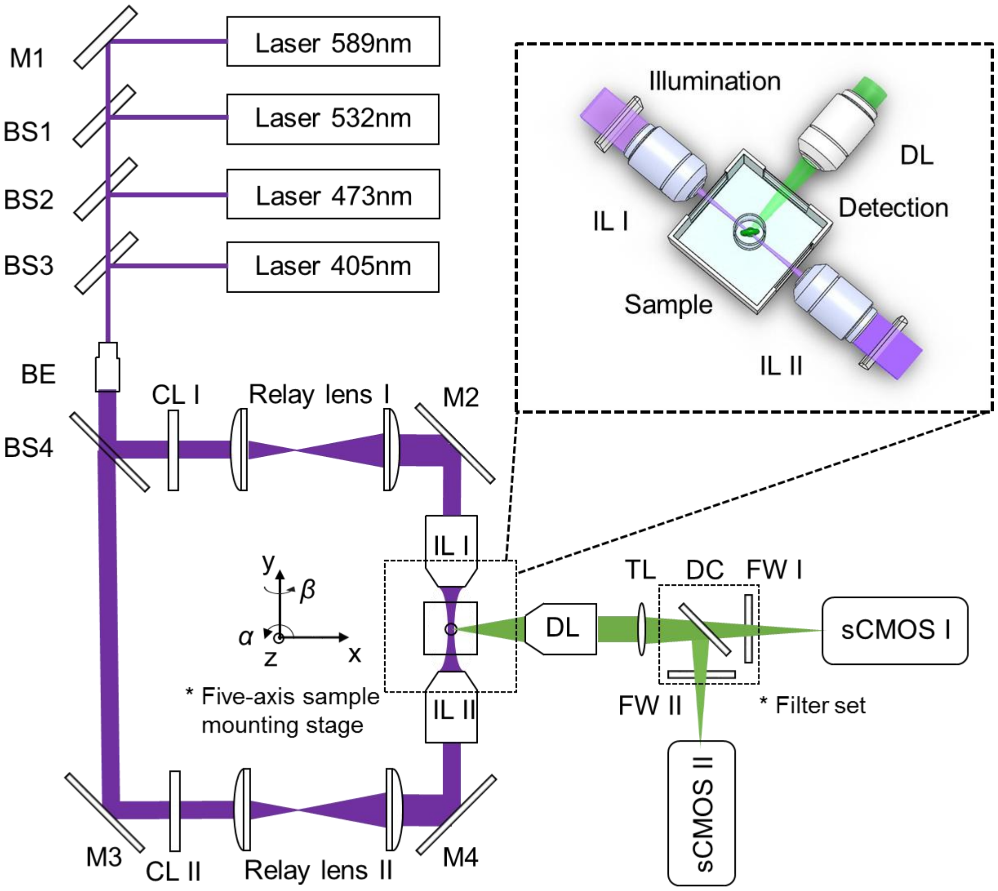

Figure S4. Schematic diagram of a dual-sided illumination and dual-channel detection for the light-sheet fluorescent microscope. M: mirror; BS: beam splitter; BE: beam expander; CL: cylindrical lens; IL: illumination lens; DL: detection lens; TL: tube lens; DC: dichroic mirror; FW: filter wheel.

Figure S5: Validation of BINS method

We treat the manual segmentation as the gold standard in this project. To validate the BINS method for zebrafish embryos, we picked up a set of data including 60 images in a single time point of the cardiac cycle and divided them into 6 groups. The dice similarity coefficient was used to measure the accuracy of the BINS method. The metric computed the number of pixels that overlap between manual and automatic segmentation results by the half the sum of the number of nonzero pixels in the two groups. The result is a value between 0 (no overlap) and 1 (perfect overlap):

$$Dice = \frac{2\times|A\cap B|}{|A|+|B|} \quad (1),$$

where A is the gold standard segmentation which in our case refers to manual segmentation, and B is the automatic segmentation mapped from the BINS method. Based on the Equation 1, the mean value and standard deviation were listed in Table S2, respectively. Overall, the mean value was 0.92, and the standard deviation was 0.02. It demonstrated the BINS method was excellent for automatic segmentation of embryonic zebrafish hearts.

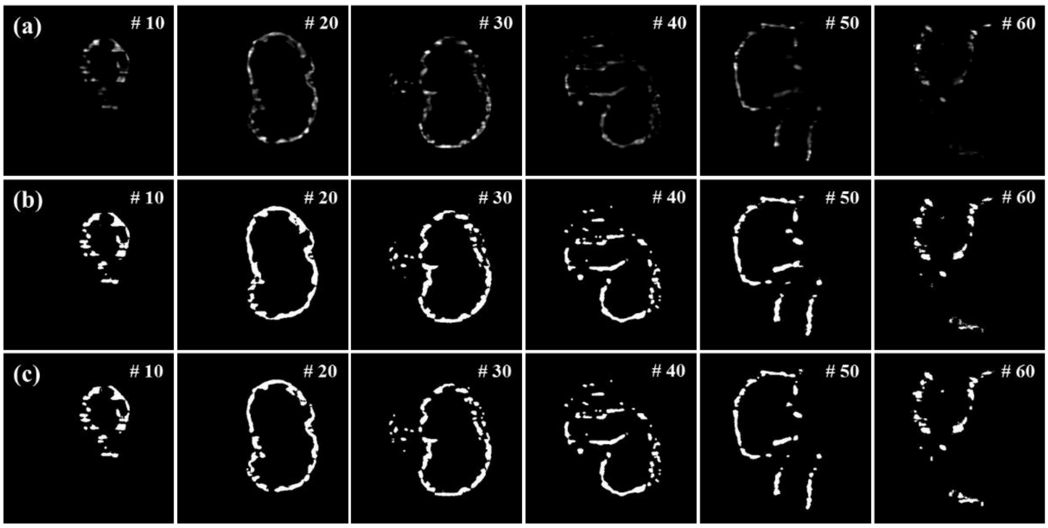

Figure S5. Representative images of (a) Raw data; (b) Manual segmentation; (c) BINS results.

Table S2

Validating results of BINS method

Slice No.	Dice similarity coefficients	
	Mean	Standard deviation
1 - 10	0.92	0.03
11 - 20	0.92	0.02
21 - 30	0.93	0.02
31 - 40	0.91	0.01
41 - 50	0.91	0.01
51 - 60	0.91	0.02

Figure S6: Refinement of polygon mesh

We need to validate the polygon mesh with the raw data, and to refine the number of vertices, edges and faces of the polygon mesh after image segmentation as well. The size of the final output is dependent on the number of vertices/edges/faces. We chose 740 x 740 x 330 voxels (~ 1200 μm x 1200 μm x 1000 μm, **Figure S6**) to highlight the essence of proper number of vertices/edges/faces. The unconstrained smoothing generated fewer vertices/edges/faces under the same condition in comparison with the constrained smoothing (**Figure S6 A-H**). For example, the number of vertices under the condition of non-compact with the smoothing extent 3 was 1.2 x 10^6, while the value of the constrained smoothing on the same premise was over 1.8 x 10^6. To acquire the smooth performance on the smartphone, we need to apply a higher level of smoothing extent and choose the compact method. For each sample, we compared the polygon mesh with the 3-D rendering data by controling over the number of vertices, edges and faces under different smoothing methods to ensure the applicability of the model for the authentic output.

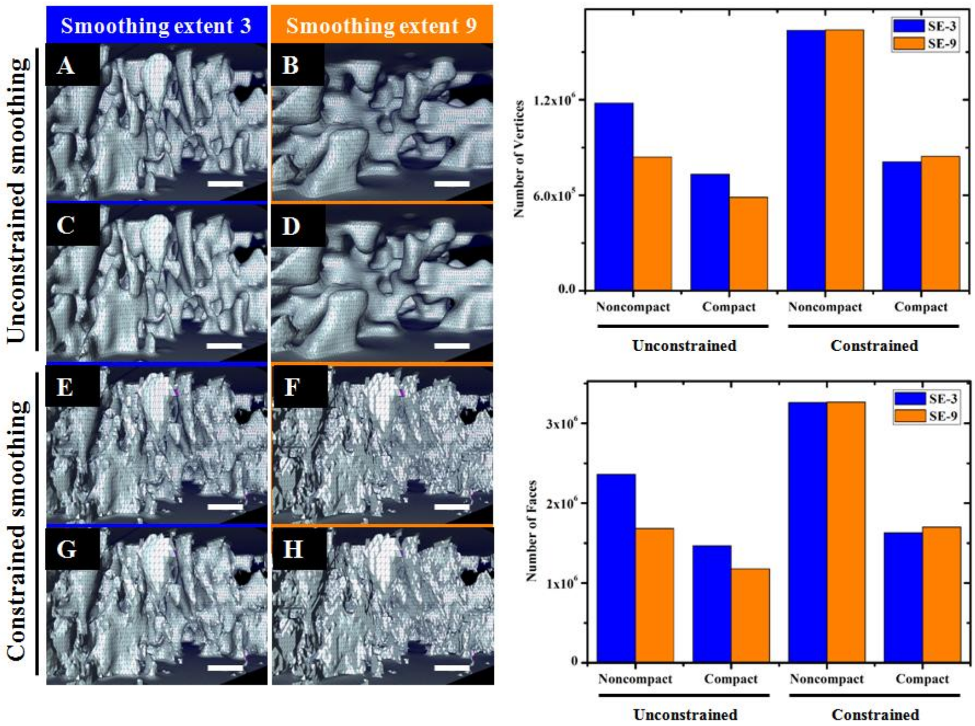

Figure S6. Comparison between the output of the unconstrained smoothing and constrained smoothing, as well as the outcome of the noncompact and compact methods in Amira. The smoothing extents are three and nine, respectively. Scale bar: 30 μm.