SUPPLEMENTAL METHODS: IMAGE PROCESSING ALGORITHM

STEP I – Image Deblurring via 3D Deconvolution

“Constrained Iterative” 3D deconvolution within Slidebook™ 5 software (based on an algorithm developed by David Agard at UCSF) was used to deblur (quantitatively reassign out-of-focus information in 3-D data while improving both axial and lateral data resolution) the original images obtained from fluorescent confocal microscopy. The difference between the original images and the images resulting from the deconvolution is shown in Supplemental Figure 1.

Supplemental Figure 1 LEFT: A sample “slice” through the middle of the thrombus prior to deblurring. RIGHT: Results of the 3D constrained iterative deconvolution applied to the same image. In both images red color is the anti-CD41 fluorescently labeled antibody that is used to mark platelet surfaces.

STEP II - Sample Movement Correction

The goal of this step is to account for the movement of the sample between each successive image capture during the in-vivo imaging. In order to do that the centroids of the black-and-white (BW) masks of each of the 2D slices are lined up with those of the rest of the 3D stack.

STEP III – Image Segmentation

The goal of this step is to create a binary representation of the thrombus structure based on the intensity signals obtained from confocal microscopy images. The image segmentation procedure is described in detail below.

Step III.a - Contrast Enhancement and Noise Removal

Although confocal microscopy attempts to eliminate all out-of-focus light, the results are by no means ideal due to equipment limitations. This is evident from Supplemental Figure 1, where dimmer out-of-focus objects appear in the acquired image even after deblurring. Since the decision criterion for elimination of the out-of-focus objects (background) is not obvious, various contrast-enhancement techniques are used in order to first exaggerate the difference between foreground and background pixels.

In particular the following contrast-enhancing operations are applied to each channel of the microscopy images:

1) Intensity histograms of each image are mapped to a new range using the Matlab “imadjust” function with a gamma correction factor of 1.5, such that the net effect is a nonlinear stretching of the histograms, weighed towards the lower (darker) output values.

2) Then, the intensity histograms of each image are transformed using contrast-limited adaptive histogram equalization (CLAHE). CLAHE operates on small regions in the image (in our case each image is broken up into 20x20 regions), such that the histogram of each output region approximately matches a user-specified histogram (in our case curved “exponential”). Artificial boundaries created by the filter between the output regions are then eliminated using bilinear interpolation. A contrast limit of 0.02 is used to avoid over saturation due to amplification of any noise that might be present in the image.
3) Finally, median filtering of the images is performed by using the “medfilt2” function in Matlab in order to remove any “speck” noise. The median filter is set to operate on a region of 2x2 pixels and each output pixel contains the median value of its’ neighborhood. Supplemental Figure 2 shows the net result of applying all three filters consequently to the red color channel of original image shown in the right hand side of Supplemental Figure 1.

Supplemental Figure 2 Result of the contrast-enhancement filters applied to the red intensity channel of the image shown in the right hand side of Supplemental Figure 1.

**Step III.b – Segmentation by Filling “Holes”**

In order to further improve the differentiation between the foreground and the out-of-focus background objects a flood-fill operation is performed on the grayscale image obtained from the previous step. The “imfill” command in Matlab is used to detect “holes” - dark areas (background) that are surrounded by lighter areas (foreground). An example of the procedure is shown in Supplemental Figure 3, where the holes are labeled with blue color. Once identified, the holes are labeled as background by setting the pixel values belonging to them to zero.

Supplemental Figure 3 Segmentation by filling “holes” in the intensity image (holes are labeled with blue color and are considered to be a part of the background).

**Step III.c – Final Binary Mask**

Finally, the 2D representation of the thrombus structure in the plane of the image is obtained by converting the intensity image to a binary one via the “im2bw” command in Matlab, with a default threshold of 0.5. This corresponds to setting all pixels that are above the midway between black and white pixel values to foreground (or ones) and everything else to background (or zeros). The final binary mask of the thrombus structure considered in the previous steps is shown in yellow color in Supplemental Figure 4.
STEP IV – 3D Reconstruction and Interpolation
The procedures outlined in the previous steps are repeated for each “slice” in the 3D stack. Since the resolution in the plane of the confocal slices is not necessarily the same as the resolution in perpendicular dimension, 3D linear interpolation is used between each 2D slice to equate the inter-slice separation distance with the inter-pixel distance in the plane of the images. This is necessary in order to generate a cubic lattice for the flow dynamics simulations using the Lattice Boltzmann method. After the interpolation, the obtained images are stacked together to form a 3D virtual representation of the thrombus structure (Left hand side of Supplemental Figure 5).

STEP V – Imaging Artifacts Correction & Validation
Since some of the fluorescent antibodies used to image the thrombus via confocal microscopy are platelet-surface markers, the 3D reconstruction of the thrombus structure could result in an imaging artifact where platelets have hollow insides. This could present a problem for porosity measurements and could also generate isolated pockets of fluid in the flow dynamics simulations, which is undesirable. In order to correct this, the “holes” inside the binary representation of the platelets are “filled” in 3D by using the Matlab BW “imfill” algorithm, where a hole is considered to be a set of void voxels that cannot be reached by filling in the background from the edge of the 3D image (analogous to the 2D holes in Step III.b).

One possible source of error in the image processing algorithm is that the procedures in Steps I & III could potentially eliminate pixels that are in-focus but are dim due to a variety of reasons (inconsistencies in light transmission through tissue, variations in antibody labeling or platelet morphology, etc). Therefore, the final thrombus geometry is validated by comparing its specific surface area (surface area-to-volume ratio) to what is expected from theory: assuming that the equivalent sphere diameter of platelets is no less than 1µm, the average specific area of a thrombus consisting of such spheres should be no greater than $S_{\text{MAX}} = 60,000 \text{ cm}^{-1}$ ($S = 3/R$, where R is the radius of the spheres). Hence, since the measured $S_{\text{ACTUAL}} \approx 24,400 \text{ cm}^{-1}$ for the thrombus is less than the theoretical upper limit, it is concluded that the reconstructed thrombus structure has not been underestimated due to the deconvolution and segmentation.

STEP VI - Thrombus Volume Estimation
In order to estimate the total volume of the thrombus for the purposes of the porosity calculation, a convex hull (Qhull) algorithm code by John D'Errico (http://www.mathworks.com/matlabcentral/fileexchange/10226-inhull) was implemented in Matlab. A Qhull for a set of points is simply the minimal convex set containing the points. In 2D it can be thought of as a rubber band wrapped around the “outside” points. The final 3D reconstruction of the thrombus and the corresponding Qhull volume are shown in Supplemental Figure 5.
An additional use of the Qhull volume is to mark the region of interest during collection of statistics within the thrombus pores. However, in order to avoid gathering data at the periphery of the thrombus structure, where exceptionally high gradients of velocities might occur, the original Qhull volume is eroded (shrunk) using the “imerode” function in Matlab. The application of an erosion operator on a binary image eliminates the boundaries of foreground regions. Mathematically it is defined as the operation: $A \Theta B = \{z \mid (B_z \subseteq A)\}$, as a result of which only the voxels locations $z$ remain, where a “structuring element” $B$ (in our case a cube of size $9^3$ voxels) translated to the location overlaps with just foreground voxels of the binary image $A$. The final eroded Qhull volume is shown in Supplemental Figure 6.

Supplemental Figure 6 Comparison of the eroded Qhull volume (Green) overlaid with the 3D thrombus reconstruction (Grayscale) in a blood vessel represented by a pipe (Red). Parts of the blood vessel have been omitted for clarity.