Supplementary Information

Programmable Control of Nanoliter Droplet Arrays using Membrane Displacement Traps

Jason Harriot,[1,2] Michael Yeh,[1,2] Mani Pabba,[3] and Don L. DeVoe[1,2,*]

[1]Department of Mechanical Engineering, University of Maryland, College Park
[2]Fishell Institute for Biomedical Devices, University of Maryland, College Park
[3]Department of Computer Science, University of Maryland, College Park

1. Supplementary Notes

Note S1: FluidScript droplet control software

The vision system software, named FluidScript, is a python program that controls on-chip pneumatic actuation, operates the camera, performs image analysis, presents a graphical user interface, and implements user-defined code defining the desired experimental steps through a separate user script. The user script has access to functions designed to facilitate sequences of ejection, capture, merging, splitting, and metering steps as needed for a particular application. The software employs a multi-threaded architecture, allowing the user script to execute blocking functions without interrupting other tasks. The full FluidScript code distribution, including example user scripts, is maintained at https://github.com/mml-umd/fluidscript. Here we outline the code structure and describe specific class methods and functionalities that can be used in custom Python scripts to perform arbitrary user-defined sequences of droplet operations. Example user scripts are included in the Github code distribution.

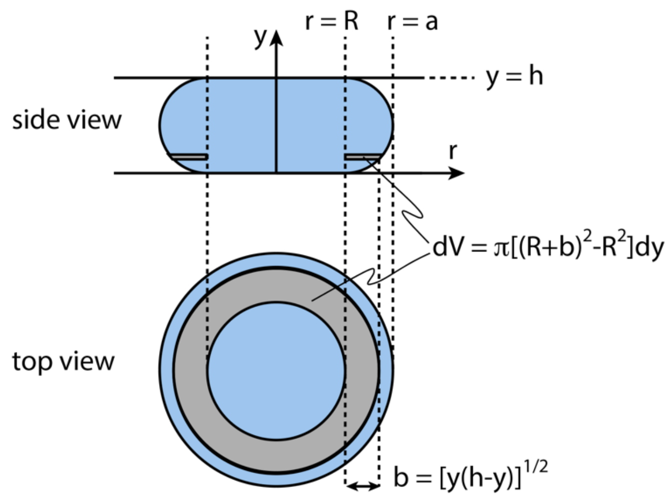
The graphical interface provides for real-time viewing and control of the experiment (Fig. 2D). Manifold valve states are shown at the top of the screen (ON in red, OFF in white), along with other diagnostic information such as frame rate, time and date, and recording status. The interface is used to define points and regions of interest (ROIs) that may be referenced within the user script for detecting droplets and triggering valves. Placing reference points and ROI elements is done by clicking, or clicking and dragging respectively. Trigger elements can be moved incrementally with the arrow keys, making alignment to microfluidic features on the device trivial. Controlling droplet operations by with trigger elements provides significant advantages in experiment setup ease and precision, as opposed to hard-coded trigger locations or stochastic operation. The interface also allows the user to control valve states manually in concert with automated script-driven operation. Sliders for brightness, contrast, and normalization filter radius are provided in a separate window. The normalization filter acts as a high-pass filter, which has the effect of reducing all large color features to middle gray, and highlighting small features, such as droplets.

The *FluidScript* class is the main class of the vision system software. This class allows the user script to pull values from slider elements in the GUI to control experimental parameters, and instantiate color channels and their corresponding slider window. Each user script must instantiate a *FluidScript* object, and define two functions that are assigned to *Fluidscript* attributes named *init* and *onFrame*. The *init* attribute is assigned to a function within the user script to be executed a single time when the script is first started from the GUI by pressing a hotkey (backspace). The selected function may be used to define the initial state of the microfluidic system, for example by setting the initial pneumatic pressure levels and defining the initial MDT actuator positions. The *onFrame* attribute is assigned to a function within the user script designed to be executed after capturing each video frame. This function is used to define the specific droplet manipulation steps to be performed, typically by employing helper functions provided in the *BlobGroup* class. This latter class includes methods for implementing the underlying work of tracking droplets in the MDT chips. Each color channel acquired from the camera has its own set of hue, saturation, and value range clamps, and exposes a set of *Blob* objects, representing droplets, to the user script through the *BlobGroup* class. Each *Blob* object contains data such as centroid position, width, length, angle, and velocity. Erosion, dilation, and minimum area sliders are also provided in the user interface to allow the user to adjust threshold settings to avoid erroneously detected blobs, if necessary. The boundaries of each blob are contracted, then expanded according to the sliders. Each color channel is instantiated within the user script, which automatically creates a corresponding window of thresholding sliders. The *BlobGroup* class provides the bulk of droplet detection, measurement, and control features which make high-level experiment design possible.

Detailed documentation for each FluidScript module and class is available in the Github repository at https://github.com/mml-umd/fluidscript/CLASS_DOCS.md.

Note S2: Droplet volume derivation.

When a droplet is constrained by the upper and lower chip surfaces, but not yet large enough to contact the channel or trap sidewalls ($h < 2a \leq$ w or $h < 2a \leq 2$R), the droplet volume may be determined analytically by integrating an annular cross-section of the droplet over the droplet height using the geometric model shown in the below figure, combined with the volume of the central droplet cylinder, to yield the result presented in Eqn. 2. In the figure, the width b of the annular disc is found by trigonometry.



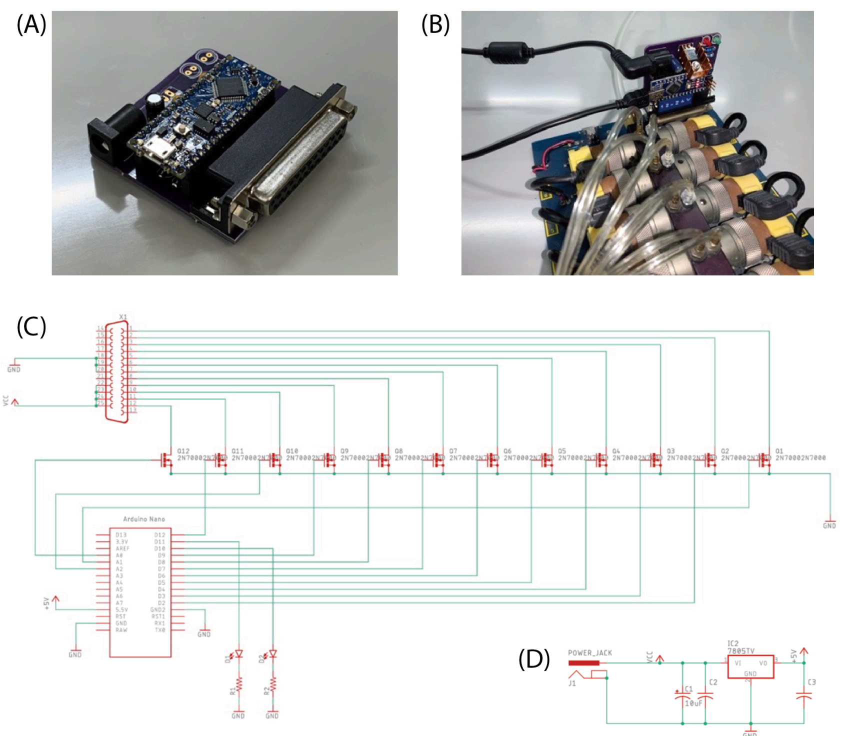
$$\begin{aligned}
V &= V_{cylinder} + V_{half-torus} \\
&= \pi R^2 h + \int_{y=0}^{h} dV \\
&= \pi h \left(a - \frac{h}{2}\right)^2 + \int_{y=0}^{h} \pi[(R+b)^2 - R^2]dy \\
&= \pi h \left(a - \frac{h}{2}\right)^2 + \pi \int_{y=0}^{h} \left[\left(\left(a - \frac{h}{2}\right) + (y(h-y))^{\frac{1}{2}}\right)^2 - \left(a - \frac{h}{2}\right)^2\right] dy \\
&= \pi h \left(a - \tfrac{h}{2}\right)^2 + \tfrac{\pi}{24}[(4 - 3\pi)h^3 + 6\pi a h^2] \\
&= \tfrac{\pi h}{24}[24a^2 + 6h(\pi - 4)a - h^2(3\pi - 10)]
\end{aligned}$$

Note S3: Pneumatic interface board hardware

Clippard pneumatic manifolds are widely used for control of microfluidic elastomer valves due to the convenience of having up to 12 valves integrated in a single manifold. To simplify interfacing with the 25-pin connector provided on the manifold, a plug-and-play interface board was developed with a regulated voltage source and Arduino nano for communication with FluidScript code running on a desktop PC. The parts list and board details are provided here. The parts list and board schematic are provided here, and a PCB CAD file is available through the Open Science Framework platform under https://osf.io/2h48b.

Parts list:

- Arduino Nano or compatible
- 15-pin female headers (x2)
- 2N7000 N-channel MOSFET, TO-92 package (x12)
- 7805 5V regulator, TO-220 package
- 220 Ω resistor (x2)
- 10 µF capacitor
- 22 pF capacitor (x2)
- 5 mm LED (x2)
- 5.5 mm DC power jack



Above image: (A) Valve manifold adapter board. (B) Adapter board seated into 25-pin manifold connector, with power and USB cables connected. (C) Control circuit schematic. (D) Voltage regulator circuit schematic.

2. Supplementary Figures

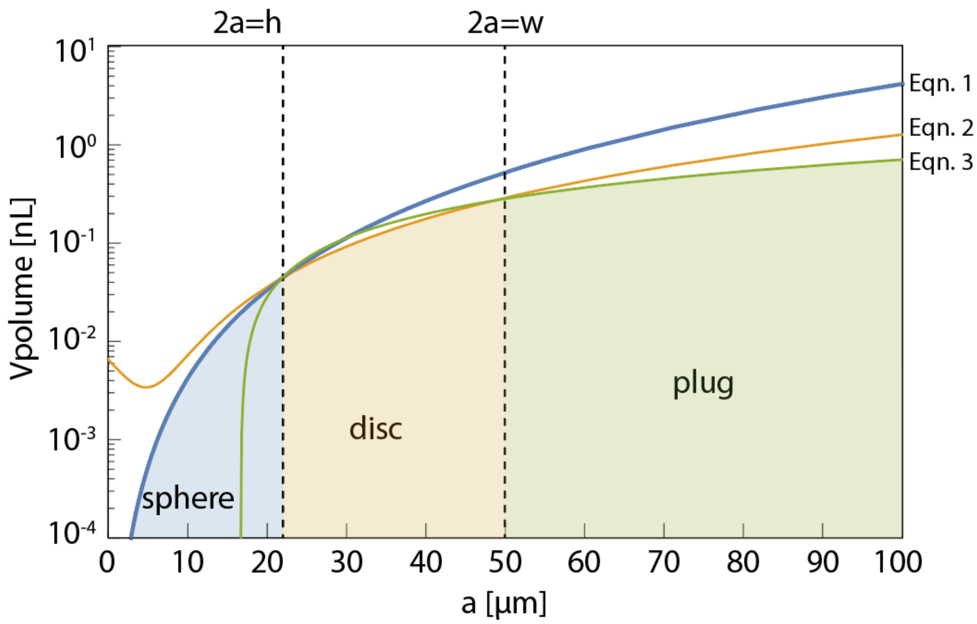


Fig. S1: Volume domains based on droplet morphology. Volume curves for small spherical droplets, intermediate circular discs, and large elongated plugs within a 100 µm wide and 40 µm tall channel. The separate regimes are presented as a function of droplet half-length or radius (*a*), and are bounded by the relevant analytic expressions in Eqns. 1-3. The same model applies for the case of a droplet in a trap, but with the upper boundary of the disc regime extended to the static trap radius.

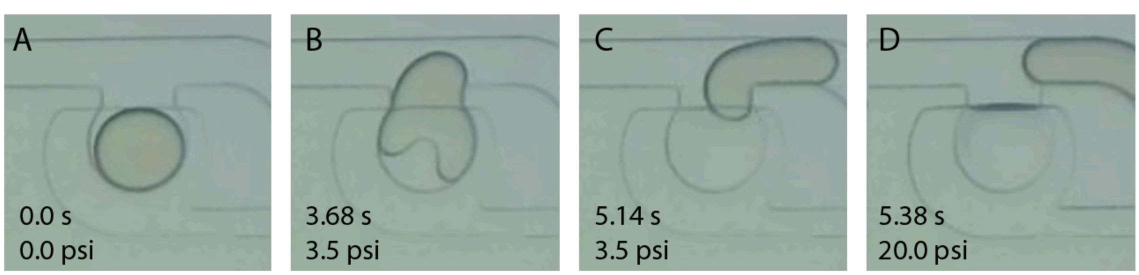


Fig. S2: Two-stage droplet ejection process. (A) A captured droplet ready for ejection. **(B)** To prevent droplet splitting when ejecting larger droplets, the pressure is first increased to 3.5 psi to partially eject the droplet. **(C)** The droplet is monitored to determine when the tail nears the trap exist. **(D)** At this point the pressure is increased to 20 psi, leading to complete droplet ejection.

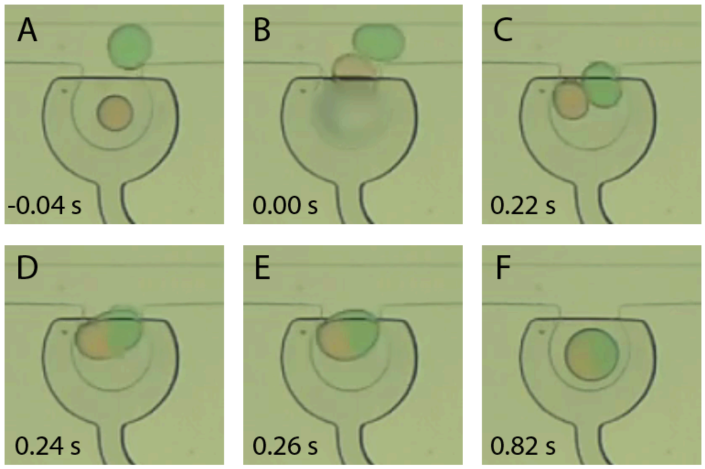


Fig. S3: Droplet merging process. (A) A green droplet approaches the target trap containing a red droplet. **(B)** When the center of the green droplet reaches the centerline of the trap (t = 0 s), the MDT control line is actuated with 2.5 psi pressure, partially ejecting the red droplet from the trap. **(C-E)** The pressure is returned to 0 psi, pulling both droplets into the trap where **(F)** droplet mixing occurs.

3. Supplementary Movies

Provided movies were captured from the filtered FluidScript video window (Fig. 2D) and display all user-defined reference points and regions of interest entered through the user interface and referenced by the script code used to control droplet operations. All movies are recorded at 4× original speed.

Movie S1. Droplet generation at increasing valve dwell times.

Movie S2. Droplet splitting by partial droplet capture.

Movie S3. Droplet splitting by oil ejection.

Movie S4. Droplet splitting by oil ejection, and child droplet capture.

Movie S5. Droplet capture and ejection steps combined with flow reversal.

Movie S6. Demonstration of repeated droplet merging events.

Movie S7. Demonstration of rapid flow reversal using the integrated H-bridge element.

Movie S8. Droplet capture, ejection, flow reversal, and merging (example 1).

Movie S9. Droplet capture, ejection, flow reversal, and merging (example 2).

Movie S10. Automated pH ladder generation via cascade dilution.

Movie S11. Droplet merging by ejection of a trapped droplet into a passing sample plug.

Movie S12. Small droplet capture and release with FC-40 as a continuous phase.

Movie S13. Large droplet capture and release with FC-40 as a continuous phase.