Automating Microfluidics: Reconfigurable Virtual Channels for Cell and Droplet Transport

Citation

Permanent link

Terms of Use

Share Your Story

ARTICLE TYPE

Automating Microfluidics: Reconfigurable Virtual Channels for Cell and Droplet Transport

Caspar Floryan,[a] Alex Nemiroski,[a] and Robert Westervelt[a,b]

The emerging field of digital microfluidics promises to solve many shortcomings of traditional continuous-flow fluidics. This technology has a few incarnations, including EWOD (eletrowetting on dielectric) and DEP (dielectrophoresis) chips. Both consist of large arrays of electrical pixels which move droplets and cells. They actuate fluids actively, have error feedback, are programmable, perform
10 operations in parallel, and do not rely on external pumps. For these reasons we foresee the increased use of digital microfluidics in the near future. We also foresee a gradual shift away from purpose-built microfluidic devices, towards multi-purpose platforms with specific applications encoded in software. To this extent we present here a new paradigm of encoding and automating microfluidic operations using video files. We use this technology to create several configurations of virtual microfluidic channels and
15 to play film clips using living cells on a DEP chip.

Introduction

Microfluidics is generating great excitement with the promise to revolutionize the modern laboratory[1,2,3,4,5]. It promises to miniaturize and automate laboratory tests, moving biology and
20 chemistry onto tiny chips. Great strides have been made towards achieving this vision. The first generation of continuous-flow, pressure driven microfluidics successfully miniaturized certain classes of applications[6,7]. These devices pump fluids through micron-scale channels, performing any number of operations
25 along the way, such as mixing, separation and chemical sensing of fluids, cells or particles. External pumps drive the fluids through these microfluidic channels. Pressure-driven microfluidics have a strong correlation between form and function, with the non-reconfigurable pattern of channels, usually
30 cast in solidified polymer, largely determining the chip's function. Several attempts have been made at designing multi-purpose pressure-driven microfluidics by implementing valves[8], abstraction layers[9], and Boolean algebra[10].

In this paper we demonstrate how moving to a digital
35 microfluidic platform[11,12,13,14] can tremendously simplify the separation of form from function, with function now completely encoded in software. We demonstrate how sample automation, implementing a channel design, and completely reprogramming a

40 *[a] School of Engineering and Applied Science, Harvard University, 29 Oxford Street, Cambridge, MA 02138, USA*
[b] Department of Physics, Harvard University, 29 Oxford Street, Cambridge, MA 02138, USA
E-mail: westervelt@seas.harvard.edu
45 † Electronic Supplementary Information (ESI) available: Video of a person dancing on the DEP chip. See DOI: 10.1039/b000000x/

chip can each be done in a matter of minutes rather than the many days required to build a new PDMS-based microfluidic device.
50 We use a digital microfluidic device and refine the concept of "virtual channels." Both of these terms will be described further in the proceding paragraphs.

Digital microfluidic chips use electromagnetic fields, as opposed to pressure-based forces, to move droplets, cells and
55 other small particles. The three main categories of this technology include DEP (dielectrophoresis) chips which move cells and droplets using dielectrophoresis[15], EWOD (electrowetting on dielectric) which modify the contact angles between droplets and the chip surface to create an actuating
60 force,[16] and magnetic chips which move cells or other objects doped with magnetic particles[17,18]. Electrodes have been designed in various sizes, ranging from micro-meters[15,19] to millimetres[13,20]. Arrays of magnetic wires and coils have been designed on the scale of 10-20μm[17,18]. They also come in
65 different shapes and configurations, ranging from 10 irregularly shaped electrodes arranged in a circle[21], to tens-of thousands of square, pixel-like electrodes arranged in a large matrix[19]. Our studies here were performed using DEP chips. DEP chips typically have smaller electrodes, in the micron range, and are
70 more attuned to manipulating cells and lipid vescicles[19]. EWOD chips traditionally have larger electrodes, in the millimetre range, and are more appropriate for manipulating larger fluid droplets.

Virtual channels are simply channels which are not bound by physical walls. Instead they trap fluids or particles using
75 electromagnetic forces and move them to a pre-determined location. This is analogous to having a room with metal balls lying on the floor and a train of magnets under the floor trapping and moving the balls along pre-determined paths.

Virtual microfluidics is an nascent field. Zhao *et al.* provide an early reference to using free surface energies inside microchannel networks to direct fluid flow.[22] More recently Basu *et al.* describe the use of Marangoni flows to confine droplets and create virtual channels.[23] Ding *et al.* presented an architecture and optimization methodology for two-dimensional electrowetting devices.[24] Most recently Dhindsa *et al.* reported a method of creating virtual channels using an array of electrowetting posts.[25] We hope to add to the state of the art be presenting a virtual microfluidics paradigm which is more versatile, more platform independent, more scalable and represents a further step up in abstraction from previously reported work.

We use digital microfluidics here to store and play a wide array of fluidic applications on a single piece of hardware using video files. We call it "video on a chip" and demonstrate its ability to run reconfigurable, virtual channels over a chip, and to play film clips using cells. All of this is done by sending GIF video files to a standard DEP Chip.

The DEP Chip

We chose the DEP chip as our hardware platform[19]. The chip is shown in figure 1. An EWOD platform could equally have been used. Our DEP chip is an integrated circuit with an array of 128 x 256 pixels on its surface. The pixels can attract and repel thousands of micron-sized cells to form distinct and highly controllable 2-D patterns. The patterns are then animated by moving cells between pixels. This is analogous to a traditional LCD or plasma display except instead of displaying colors, the DEP array displays a binary state of cell or no-cell. The pixels are approximately the same size as the cells, 11x11μm in size.

Cells are attracted to pixels based on their voltage. Each pixel can be individually potentiated to either 0 or 5V, resulting in a complex pattern of electrical fields over the chip's surface. Our chip is run at 1MHz where cells are attracted by a dielectrophoretic force to regions of highest electric field. This force is described by equations (1) and (2), and is a function of the particle's dielectric constant ε_p, the surrounding medium's dielectric constant ε_m, the magnitude of the electric field E_{RMS}, and the frequency of the field ω. Our device produces maximum electric fields of 50kV/m. Cells are not harmed in fields below 100kV/m[26].

$$F_{DEP}(\omega) = 2\pi\varepsilon_m r^3 CM(\omega)\nabla E_{RMS}^2 \qquad (1)$$

$$CM(\omega) = Re\left[\frac{\varepsilon_p - \varepsilon_m}{\varepsilon_p - 2\varepsilon_m}\right] \qquad (2)$$

The integrated circuit[28] is made with a 0.35μm gate length CMOS process with four metal layers and 5 V transistors. Each pixel consists of three circuit blocks: (1) The static random access memory (SRAM) to store the state of the pixel, (2) Control transistors that allow an RF voltage to pass depending on the state of the SRAM, and (3) drive transistors to pull up or pull down the capacitive load of the pixel. The RF voltage is sourced off the chip and can range from 20Hz to 1.8MHz. The time taken for the pixel to ramp up or down is short compared to the period of the RF voltage, with an RC time constant under 1ns.

To individually address the 256 x 128 pixels, a decoder identifies each row with a unique 8-digit binary number. The state of each of the 128 pixels is sequentially loaded into a shift register. The shift register is updated using a 2-phase clocking scheme. Control signals determine whether bits in a row are written or read from the shift register.[19]
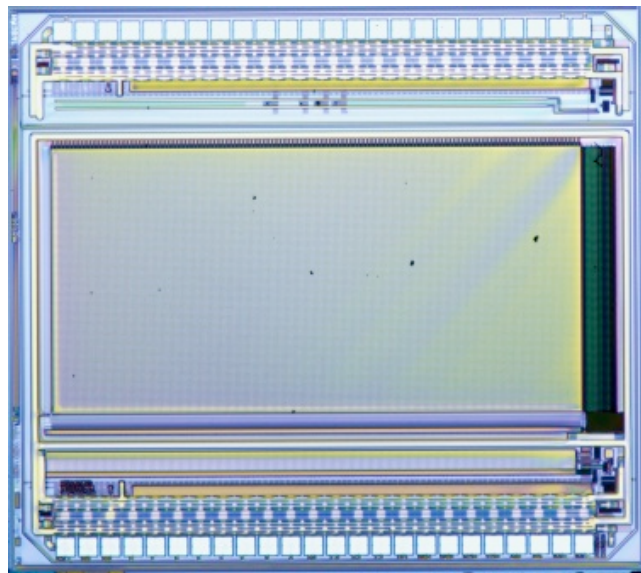

Fig. 1 The DEP chip contains an array of 128 x 256 individually-addressable pixels on its surface. Each pixel is approximately 11 x 11 μm and the entire chip is 2.32 x 3.27mm in size. Each pixel can be potentiated to either 0 or 5V. Cells or droplets are dielectrophoretically attracted to regions of highest electric field gradient.

Methods

Playing videos on a DEP chip is a simple process. We will begin by describing in more detail how cells move over our chip and follow this with a discussion about the structure of our videos. We'll also describe the scripts we've developed to accelerate translating a channel design into a video file and end by addressing issues of cell adhesion to the surface of our chip.
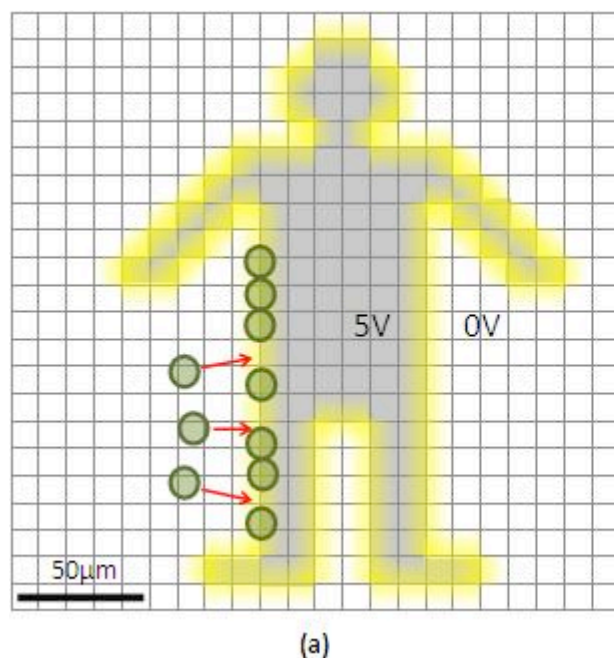
We can activate any combination of pixels on our chip. The active pixels assume a different voltage from the rest of the chip, creating localized electric fields. Cells are attracted to these areas of highest electric field. The electric fields can be dragged from pixel-to-pixel, moving cells along with them. The CMOS circuitry built under each pixel controls whether it is activated. It controls the sign of an external RF source, reversing its sign to activate the pixel. Active pixels are thus always out-of-phase with the rest of the chip. The RF source is a 1MHz square wave, an optimum frequency for the dielectrophoresis of cells.

Figure 2 (a) depicts one frame from a video of a dancing person being displayed on our chip. Each square represents an 11x11 μm pixel, with active pixels being out of phase with clear ones. The active pixels, shaded in figure 2 (a), are arranged in the shape of a person but can also be arranged to form any other arbitrary shape. An alternating electric field is produced where the phases change, depicted as a shaded outline of the person.

This electric field attracts cells or other particles that have a dielectric contrast with the surrounding fluid. Observing the chip through a microscope, one then sees an outline of a person formed entirely from cells. Our experiments are performed using both live yeast cells (Saccharomyces cerevisiae) and live ED19 mammalian cells. Frames with the person in slightly different positions are each sequentially loaded onto the chip. With each frame transition, the cells follow the person's outline.

Our videos consist of a sequence of black-and-white frames stacked together in a GIF file. The GIF file is read by our MATLAB control software. The software stores each frame individually in a binary matrix with in and out-of-phase pixels represented by 1's and 0's. Each matrix is then individually sent to the chip with a delay time between matrices specified by the user. The delay time is chosen to hold on to as many cells between frames – some cells will float away as the dancer moves – while maintaining the video's fluidity. The chip is connected to our computer through a USB cable.

Figure 2 (b) depicts a train of active pixels moving over the chip. These pixels trap and move cells to form a microfluidic "virtual channel." Figure 2 (b) depicts a short linear channel but more complicated configurations are also made. A channel can make 90-degree turns, channels can be merged or bifurcated, and many channels can be combined to created complicated configurations. A tree junction configuration is shown later in figure 4 (b).

A novel aspect of these virtual channels is that they are completely encoded in GIF video files. Making a channel on the chip is as simple as playing the video file on a computer. After one channel configuration has served its utility, the user merely has to play a different GIF file to completely reconfigure the device. This process of changing configurations takes seconds as opposed to the days required to build a new configuration in a traditional microfluidic device.

The GIF files of virtual channels are played identically to those of the dancer in figure 2 (a). The GIF file is read by our MATLAB control software. The software then translates each frame into a matrix and sends the matrices sequentially to the chip. The delay time between each frame is set by the user in a way that balances speed and the cell's affinity to stay trapped.
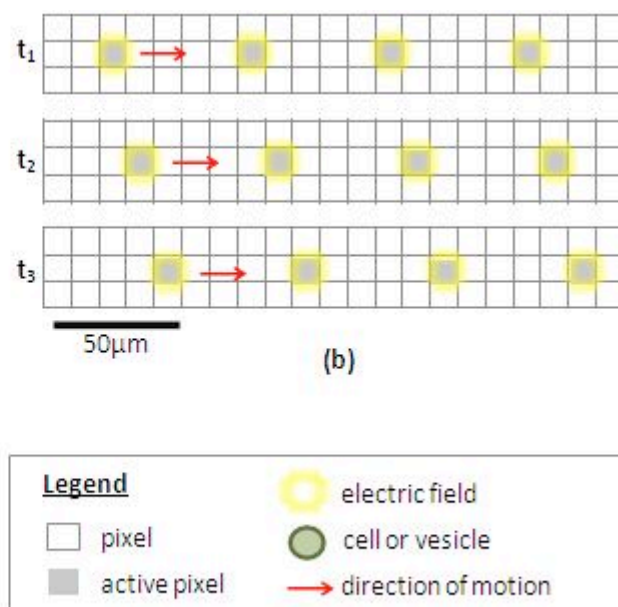

Fig. 1 (a) A diagram of the chip's surface with pixels activated in the shape of a person. Each square represents one 11x11μm pixel, with the gray-shaded pixels being π out of phase with the clear pixels. Electric fields are produced at the border between in-phase and out-of-phase pixels, here represented by the yellow shaded regions. Cells, vesicles or other particles are attracted to electric field maxima and cumulatively form the person's outline. A sequence of images of the person dancing is sent to the chip resulting in cells following the person's changing outline. (b) A train of active pixels moves left along the chip. The positions of the pixels are indicated at times t_1, t_2 and t_3, moving one step left at each time interval. The active pixels form a conveyor belt which traps and moves cells, vesicles or other particles. Many such conveyor belts are arranged together to form a network of "virtual channels." These virtual channels are created by simply playing a GIF video on the chip.

The video of the dancing person, as depicted in figure 2 (a), is created by first capturing a video from YouTube using any number of applications which save streaming videos, such as KeepVid, ClipNabber and Savevid. We used the application KeepVid. The video must next be formatted for playback on our chip. It has to be transformed into a binary (black-and-white) format and be resized to 128x256 pixels to match the dimensions of our chip. Individual frames are extracted from the video using Adobe Photoshop CS4. The color channels of each frame (Red, Green, Blue) are merged to form greyscale images. Each frame is then thresholded such that pixels below a chosen greyscale value become black and the remaining pixels turn white. The frames are then resized to 128x256 pixels. This entire process from extracting frames to resizing and saving is automated in Adobe Photoshop and took a few minutes for the videos we worked with. The black-and-white frames are then reassembled into a video. This is done using a MATLAB script which stacks

frames and saves them in GIF video format.

These reformatted GIF videos are played on the chip using a personal computer with control software written in MATLAB. The software opens a GIF file and extracts each frame into a
5 matrix. It sends each frame individually to the chip with a delay time between frames that can be controlled by the user.

The virtual channels depicted in figure 2 (b) are created by drawing lines representing channels in a graphics software, such as Paint, Adobe Photoshop, or Paintbrush. We used Windows
10 Paint. A linear channel is simply represented by drawing one straight line in Paint. We then developed a MATLAB module which reads this Paint image and produces a sequence of frames, each containing a train of black pixels, with the pixels moving forward one step with each frame as shown in figure 2 (b). This
15 MATLAB module is capable of determining the start points of channels, following channels through bends, and merging and bifurcating them. The module then can play the frames directly on the chip or saves a GIF video for future playback. We have thus reduced the entire process of designing and implementing
20 virtual microfluidic channels to sketching them in Paint.

In order to enable a conventional PC to control the DEP chip we designed a plug-n-play controller capable of interfacing the DEP chip through a standard USB port to a PC or Mac. The control hardware is packaged in a small 7.5"x5.5"x1.5" box that
25 uses a Texas Instruments TMS320F2812 microcontroller with internal SRAM capable of storing 512 DEP chip frames and an FT245RL USB interface chip for PC communication.

The controller runs a simple command interpreter, allowing the USB host to control the DEP chip with a series of ASCII
30 commands over a USB Virtual COM Port. To load a the frame of video to the DEP chip, MATLAB first parses the data as described above, then transfers it to the DEP controller's internal SRAM by sending a single ASCII command followed by frame data. Next, another command loads the frame data to the DEP
35 chip's integrated shift register, and the frame is displayed.

The microcontroller also has a dedicated clock output used to generate the DEP chip's 5Vpp RF trapping signal. A single command allows us to set the DEP trapping frequency in a range of 20Hz - 10 MHz. Other commands control the timing and
40 sequence in which frames stored in the controller memory are loaded onto the DEP chip. These commands can be initiated individually by the PC, or concatenated into a script and run automatically by the controller.

The microcontroller updates the DEP chip at a refresh rate of
45 30Hz, suitable for streaming video. This time is limited by the 1MHz clock driving the chip's shift register, and leads to a total time of 33ms to load each of the 128x256 pixels of a single frame onto the DEP chip.

Our proof-of-concept experiments were performed using live
50 yeast cells in a 5%w/w mannitol solution. The solution is isotonic with the yeast cells, preventing osmotic pressures from stressing the cell membranes. A drop of cell solution was pipetted onto the chip and covered with a microscope cover slip. The chip was then activated and videos could be played
55 immediately. Pluronic F-127 was also added to the solution to prevent cells from adhering to the chip. The F-127 inhibits cell-surface protein absorption and is a previously cited compound used for digital microfluidics.[27]

Results

60 We successfully played two types of videos on our chip. The first was a video of a dancer and the second was a video of virtual microfluidic channels. The video of the dancer shows an action or story being executed, using live cells as moveable "ink." The video of virtual channels is a more practical application of video-
65 on-a-chip, with its primary purpose being to automate the transport of micron-sized objects.

Figure 3 shows a video of the dancer being played on our chip. The dancer was originally a short segment of a commercial, which featured silhouettes of dancers in front of a colored
70 background. We successfully converted and resized this video using Adobe Photoshop to a black-and-white, 128x256, GIF format for replay on our chip. The color and size conversion was done manually for the first frame and then automated using Adobe Photoshop for the remaining frames. A MATLAB script
75 was then used to cascade all the frames into one GIF file. A solution of live yeast cells was placed on the chip's surface and the GIF file was sent to the chip. In figure 3, pixels with yeast cells appear dark and empty pixels appear bright. Each dark dot represents a single yeast cell. Screenshots were taken at five
80 second intervals. The video starts with cells randomly distributed over the surface of the chip. Once the GIF file starts playing, cells are attracted by electric field maxima to the dancer's outline and follow with the outline as it moves from frame to frame. The dancer's outline sweeps throughout the chip, collecting ever more
85 cells. This is observed with the outline becoming darker and thicker in the later frames. Occasionally the dancer made a quick move and cells were unable to follow quickly enough. This resulted in some cells being lost.
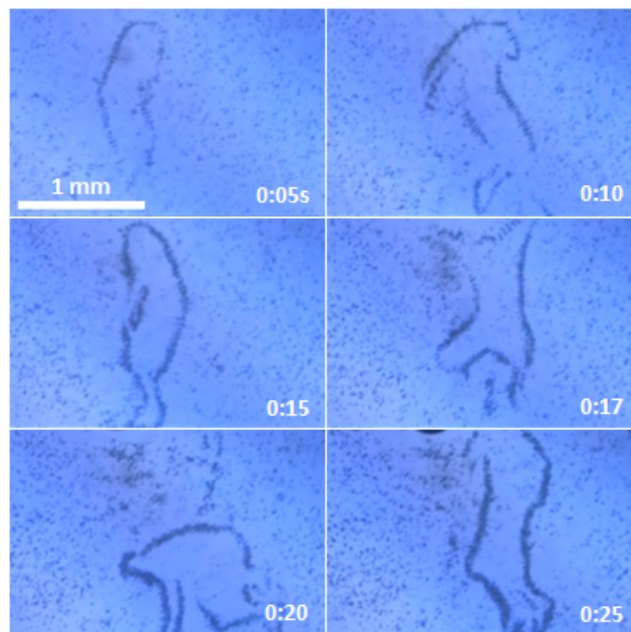

90 **Fig. 3** Screenshots from a video being played with live yeast cells. This was originally a short segment from an iPod commercial, reformatted to play on our chip. Cells are attracted to the active area around the dancer's outline and follow along from frame-to-frame. The dancer is approximately 2 mm tall and frames were collected every five seconds,
95 with one additional frame at 0:17 to show the dancer jumping.

The original version of the video shown in figure 3 was played at 30 frames per second. Photoshop was used to extract every fifth frame, or six per second. These frames were then stacked into a GIF file and played on the chip using the MATLAB control
5 software at a rate of 2 per second. The video on the chip was thus played at 1/3rd of the original speed and using 1/5th the number of frames. We used a reduced number of frames to speed up the frame processing time in Photoshop and we played them at a reduced rate to increase the cell's affinity to the dancer between
10 frames. Enough frames had to be retained so that the dancer didn't make large jumps in position between frames. Cells would not be capable of following large jumps between frames.

We also used a video-on-a-chip to create linear microfluidic channels and a tree junction. The linear channels are seen in
15 figure 4 (a). Three parallel channels are shown carrying discrete clusters of cells. Each cluster of cells corresponds to a single set of active pixels. The cells follow along as the active regions move down from pixel to pixel. The direction of movement is shown with the white arrows. We had full control over the
20 spacing between active regions, their speed and their size. Smaller active regions, analogous to narrower fluidic channels, carried fewer cells. Similarly, larger active regions carried more cells and were analogous to wider channels.

The tree junction is shown in figure 4 (b). It consists of a
25 channel which bifurcates into several channels. Our implementation of the tree junction splits cells equally between channels. Cells are also seen being carried in clusters. At junction points, the cluster is split in two with some cells going left and the rest going right. At each junction a binary decision
30 could be made about the direction in which a cell or droplet will proceed. For example, if a cell is red it could proceeds to the right and if it is blue it proceeds left. Figure 4 (b) shows closer cluster spacing than the linear channel implementation in figure 4 (a).
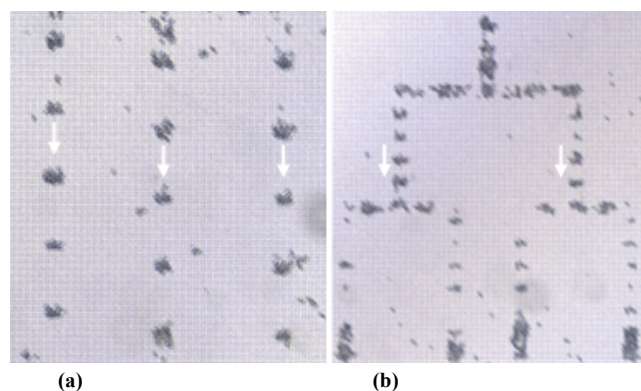
35

(a)   (b)

Fig. 4 Our implementation of microfluidic virtual channels. A train of active pixels traps and moves cells over the surface of our device. The
40 cells are clumped into discrete bundles above the active pixels. (b) A virtual tree junction separates a channel into four streams. Our implementation currently separates the cells evenly into the four streams but future improvements with capacitative sensing may be capable of actively separating the cells into different channels based on their intrinsic
45 properties.

Conclusions

We have shifted microfluidic functions from hardware to software using the concept of video-on-a-chip. We used videos
50 to do two things: to create virtual channels which carry cells and droplets and to play film clips. A DEP chip was used as the hardware platform and the ubiquitous and highly interchangeable GIF format was used for videos. The concept of "Video-on-a-chip" can also be applied to EWOD devices which contain large
55 arrays of electrical elements. We imagine our work here as a step towards, and as one possible incarnation of hardware-independent, general purpose microfluidic devices. We also see digital microfluidics gaining further acceptance and use in the lab-on-a-chip community.

References

[1] C. Kumar, editor. *Microfluidic Devices in Nanotechnology.* Wiley; Hoboken, New Jersey: 2010.
[2] K. Herold and A. Rasooly, editors. *Lab on a Chip Technology Volume 2: Biomolecular Separation and Analysis*. Caister Academic Press; Norfolk, UK: 2009.
[3] P. Dittrich and A. Manz, *Nature Reviews Drug Discovery*, 2006, **5**, 210-218
[4] H. Andersson and A. Van den Berg, *Sensors and Actuators B:Chemical*, 2003, **92**, 315-325
[5] S. Mouradian, *Current Opinion in Chemical Biology*, 2002, **6**, 51-56
[6] J. Melin and S. Quake, *Annual Reviews of Biophysics and Biomolecular Structure,* 2007, **36**, 213-231
[7] T. Squires and S. Quake, *Rev. Mod. Phys.*, 2005, **77**, 977-1026
[8] M. Unger, H.P. Chou, T. Thorsen, A. Scherer and S. Quake, *Science*, 2000, **288**, 113-116
[9] W. Thies, J.P. Urbanski, T. Thorsen and S. Amarasinghe, *Natural Computing*, 2008, **7**, 255-275.
[10] M. Prakash and N. Gershenfeld, *Science*, 2007, **315**, 832-835
[11] H. Lee, D. Ham, R. Westervelt, editors. *CMOS Biotechnology*. Springer; New York: 2007.
[12] R. Fair, *Microfluidics and Nanofluidics*, 2007, **3**, 245-281
[13] I. Barbulovic, H. Yang, P. Park and A. Wheeler, *Lab on a Chip 8*, 519-526 (2008)
[14] S.K. Chung, K. Rhee and S.K. Cho, *International Journal of Precision Engineering and Manufacturing 11*, 991-1006 (2010)
[15] N. Manaresi, A. Romani, G. Medoro, L. Altomare, A. Leonardi, M. Tartagni and R. Guerrieri, *IEEE Journal of Solid-State Circuits*, 2003, **38**, 2297-2305
[16] M Pollack, R. Fair and A. Shenderov, *Applied Physics Letters*, 2000, **77**, 1725-1726
[17] H. Lee, A. Purdon, V. Chu and R. Westervelt, *Nano Letters*, 2004, **4**, 995-998

18 H. Lee, Y. Liu, R. Westervelt and D. Ham, *IEEE Journal of Solid-State Circuits*, 2006, **41**, 1471-1480

19 T. Hunt, D. Issadore and R. Westervelt, *Lab on a Chip*, 2008, **8**, 81-87

20 Vijay Srinivasan, Vamsee K. Pamula and Richard B. Fair, *Lab on a Chip*, 2004, **4**, 310-315

21 B.N. Hsu, *Digital Microfluidics*, 2010 [Online]. Available: http://microfluidics.ee.duke.edu/. [2010, Dec 12].

22 B. Zhao, J. Moore and D. Beebe, *Science 291*, 1029-1026 (2001)

23 A. Basu and Y. Gianchandani, *J. Micromech. Microeng.*, 2008, **18**, 115031

24 J. Ding, K. Chakrabarty, R. Fair, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2001, **20**, 1463-1468

25 M. Dhindsa, J. Heikenfeld, S. Kwon, J. Park, P. Rack and I. Papautsky, *Lab Chip*, 2010, **10**, 832-836

26 T. Hunt. *Integrated Circuit / Microfluidic Chips for Dielectric Manipulation*. PhD Thesis. Harvard University, Cambridge MA, 2007. Print.

27 V. Lu**k,** G. M**o,** and A. Wheeler, *Langmuir*, 2008, **24**, 6382-6389