Title

Design Automation of Paper Microfluidic Devices

Author

Potter, Joshua

Publication Date

2022

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Design Automation of Paper Microfluidic Devices

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Joshua Matthew Potter

June 2022

Dissertation Committee:

Professor Philip Brisk, Chairperson
Professor William H. Grover
Professor Victor Rodgers
Professor Craig Schroeder
Professor Daniel Wong

The Dissertation of Joshua Matthew Potter is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to thank Dr. Philip Brisk, my advisor, mentor, advocate, and colleague whose support and guidance was a lode stone and a bulwark throughout a non-traditional and turbulent pursuit of higher education that encompassed political turmoil, educational tribulations, and several variants of a global pandemic that highlighted the need for the research we were undertaking. While initially a source of funding for my education, the paper microfluidic device automation project introduced to me by Dr. Brisk, became the intellectual challenge that ultimately led me to pivot from embedded systems research to microfluidics and a academic and professional path traversing the spaces between Computer Science and Bioengineering, ultimately providing me a path to a future of which I had never conceived.

The following works have been previously published and contributed significantly to this dissertation and were supported by the National Science Foundation Division of Computing and Communication Foundations under grants 1423414, 1351115, and 1536026, and the National Science Foundation Division of Industrial Innovation and Partnerships under grant 1640757:

- Potter, J., Grover, W. H., & Brisk, P. (2017, May). Design Automation for Paper Microfluidics with Passive Flow Substrates. In Proceedings of the on Great Lakes Symposium on VLSI 2017 (pp. 215-220).
- Potter, J., Grover, W. H., & Brisk, P. (2020). Dynamic Radial Placement and Routing in Paper Microfluidics. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

- Potter, J., Brisk, P., & Grover, W. H. (2019). Using printer ink color to control the behavior of paper microfluidics. Lab on a Chip, 19(11), 2000-2008.

- Potter, J., Brisk, P., & Grover, W. H. (2021). Rapid development and optimization of paper microfluidic designs using software automation. Analytica Chimica Acta, 1184, 338985.

This dissertation is dedicated to my wife Sheila for her heroic support and stoic advocacy in my academic pursuits and my son Zachariah whose growth during my tenure provided the grounding and milestones for my efforts.

Additional gratitude to my mother Dawn, who provided me with my first personal computer and unwavering belief in my abilities.

In memory of my father Bruce (1951-2019),
who passed before he could see my program completed.

ABSTRACT OF THE DISSERTATION

Design Automation of Paper Microfluidic Devices

by

Joshua Matthew Potter

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, June 2022
Professor Philip Brisk, Chairperson

The emerging demands for healthcare where access is limited due to political, environmental, or socio-economic factors have been driving research into bio-medical devices that perform in both diagnostic and therapeutic roles at lower costs and greater accessibility. Paper microfluidic devices are used in many applications, particularly medical diagnostics and offer an excellent combination of utility and low cost making them particularly valuable in resource-limited applications and point-of-care usage across a wide variety environmental conditions. Microfluidic biological diagnostics continue to mature as researchers discover new ways to exploit the technological possibilities, and address liabilities. The increasing complexity of paper-based microfluidic devices beyond home pregnancy tests is driving the need to produce new tools and methodologies that enable more robust biological diagnostics and potential therapeutic applications. However, the process of developing new paper microfluidic devices is limited due to having to manually design and fabricate designs to research. Often, researchers must design scores of different devices to find a combination of parameters that functions as expected.

In this work, a novel software framework to support automated development of paper-based microfluidic devices is introduced to facilitate both research and fabrication to accelerate the investigative process and reduce material utilization and manpower. Unlike to existing lab-on-a-chip technologies, paper-based microfluidics differs in terms of substrate technologies and use a passive flow method to deliver fluids and reagents for assays. While numerous analogies between microfluidics and semiconductor technologies have been espoused, the physical differences between the fluid dynamics and electrical current are significant which suggests that current trends in physical design for microfluidics must change course in order to be of practical use to designers. Within this framework, a methodology is introduced to address design automation such as dynamically placing and routing microfluidic components in a non-discrete design space while avoiding invalid design layouts, accounting for fluid volume usage, surface area utilization, and the timing required to perform specified biological assays and also optimizing device parameters, enabling researchers to focus on the science and thereby accelerating the development of new, low-resource paper microfluidic devices for a developing world.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

The gap in healthcare accessibility continues to widen as negative health outcomes are rising in all strata where resources are scare, economic and political systems are weak, and/or financial means and incentives directly oppose the needs of the populace. Along the spectrum of therapies from easily treatable conditions up to potentially life threatening diseases seen each year, approximately 5 million people worldwide die from AIDS and tuberculosis, another 4.3 million die from respiratory infections, around 2.9 million die from enteric infections, and about 1 million die from malaria. Virtually all of these deaths occur in developing countries [68] that may not have necessary economic or logistical resources to provide adequate healthcare to their populations. Those regions subjected to dysfunctional political systems or social upheaval also have the additional obstacles of sporadic access to the most basic necessities for everyday survival. Furthermore, the threat of a global pandemic on the scale of tens of millions of people infected and deceased like the 1918

Spanish flu [78] or a strain of influenza like H1N1 [30] has been of concern for nearly 100 years and the COVID-19 pandemic underscores this need for rapid, inexpensive, and widely-available medical diagnostics.

Most of these diseases can be successfully managed or cured if they are diagnosed in time, but developing diagnostics for use in resource-limited settings is challenging. Diagnostics suitable for use in resource-limited settings have the potential to save millions of lives each year and improve the quality of life worldwide. Even in first-world countries – where financial incentives in the marketplace directly oppose the needs of the populace – a lack of adequate diagnostics makes healthcare less efficient and a financial burden to society. Better point-of-care diagnostics can play a crucial role in health care by providing doctors with rapid diagnoses, enabling treatment to begin while the patient is still at the hospital. This reduces the number of hospital visits and helps patients recover faster; however, developing diagnostics for successful use in resource-limited and point-of-care settings is a formidable challenge. Consequently, there is an urgent need to develop and provide both diagnostic and treatment technologies that can improve healthcare for those who live in these areas.

Meanwhile, efforts to identify and mitigate global threats of diseases [81, 26] are continually being developed and deployed wherever resources are limited and needs are greatest. In this context, many researchers have designed and developed a wide variety of microfluidic diagnostic devices that can reduce materials costs while lowering the volumes of reagents and fluids needed for testing [94, 33, 89]. Moreover, they have the potential to enable doctors and researchers to perform diagnostics and provide treatments while in the field, and to reduce the need for time-consuming formal laboratory work. These devices may enable doctors to provide rapid diagnoses and to enable treatment while patients are still at the hospital, reducing the number of visits and speeding up recovery times [118]. However, these technologies are too expensive and cumbersome to make a significant impact in the developing world [98].

The *World Health Organization (WHO)* has determined a set of characteristics essential for diagnostic and point-of-care applications in resource-limited settings [68] known as the *ASSURED* criteria: (Fig. 1.1)

Affordable by those at risk of infection
Sensitive *(few false-negatives)*
Specific *(few false-positives)*
User-friendly *(requiring minimal training)*
Rapid *(treatment at first visit)* and **R**obust *(no refrigerated storage)*
Equipment-free *(no additional equipment needed for use)*
Delivered to those who need it *(small and portable)*

Figure 1.1: The World Health Organization's ASSURED Criteria

No single diagnostic technology satisfies all of these criteria, but *paper microfluidic devices* (PMD's) may come the closest [74]. While other microfluidic technologies directly actuate fluid transport via pressure [103] or electrical current [83], paper-based devices are passive, cheap to produce, and disposable making them attractive for potential consumer products like simple lateral-flow home pregnancy test, perhaps the most common paper microfluidic device. These advantages are why paper microfluidic devices are currently being used to detect COVID-19 antibodies in blood to determine if a patient was previously infected by the novel coronavirus SARS-CoV-2 [6]. Much more complicated paper microfluidics can also be developed that include complex networks of paper channels,[72, 73] paper-based valves and other flow-control structures,[11, 14] and even origami-inspired designs.[62, 51].

Low-cost and easy-to-use diagnostic technologies have the potential to positively impact healthcare outcomes in these situations, and many researchers look to microfluidic technologies for its potential to achieve that impact [108]. Microfluidic devices can reduce costs of materials through reducing their size in construction as well as lowering the volumes of reagents and fluids needed for testing. Moreover, they have the potential to enable doctors and researchers to perform diagnostic and treatments while in the field and reducing the need for time-consuming formal laboratory work as well as cutting the time to treatment and recovery of patients. In contrast to expensive conventional laboratory-scale instruments or integrated laboratories-on-a-chip, paper microfluidic devices are made from inexpensive materials, are easily mass-produced, and exhibit the high sensitivity and specificity that are hallmarks of more complicated microfluidic technologies. While other microfluidic technologies directly actuate fluid transport via pressure [103] or electrical current [83], application specific, paper-based microfluidic devices are passive, cheap to produce, and disposable. Meanwhile their simple operation is user-friendly and easy to use and have the potential to directly impact the everyday healthcare of any population.

Within this context, many researchers have designed and developed a wide variety of microfluidic diagnostic devices that can reduce costs through miniaturization and automation. Low-cost and easy-to-use diagnostic technologies have the potential to positively impact healthcare outcomes, especially in developing countries and in impoverished communities. These devices enable doctors to provide rapid diagnoses and to enable treatment while patients are still at the hospital or even in the field potentially reducing the number of visits and speeding up recovery times. While significantly beneficial for persons living in the developed world, many of these cutting-edge technologies are too expensive and cumbersome to make a significant impact in the developing world, especially in remote areas. As a result, negative health outcomes are rising in all strata that limits or otherwise fractures access to cost-effective healthcare. There is an urgent need to develop and provide both diagnostic and treatment technologies that can address and improve the healthcare for the disaffected groups burdened with obstacles to care that the more affluent groups do not have.

1.2 Paper Microfluidic Application Devices

Paper microfluidic or lateral flow devices have several advantages over traditional lab-based analyses, including low materials cost, ease of use, and little or no additional equipment needed [42, 65, 22]. In a simple but powerful example, the home pregnancy test consists of a single strip of paper; when one end of the paper is immersed in urine, capillary action drives the urine past antibodies specific to human chorionic gonadotropin (hCG), a protein present in the urine of pregnant women. If hCG is present in the urine sample, it participates in a set of binding reactions that culminate in a color change at the opposite end of the paper (and therefore a confirmed pregnancy diagnosis) [13]. In a much more complex example (Fig. 3.3), the fluidic multiplexer device [72],[86] is a multi-layer, multi-channel device that allows the end user the ability to route fluid from a single source to two or more sinks after the device has been fabricated.

Paper microfluidic devices use paper and capillary action (wicking) to drive samples, reagents, and other fluids through a series of processing and analysis steps inside porous channels. They requires no external valves or pumps to control it; thus significantly reducing the cost and size of paper microfluidic devices compared to traditional laboratory diagnostic instruments. Using dehydrated reagents immobilized in the paper eliminates the need for refrigeration and making them more robust than laboratories-on-a-chip or laboratory-scale instruments. Being small and disposable, paper microfluidic devices can easily be delivered to doctors in the field and stocked at the point-of-care. As such, these devices are poised to proliferate as healthcare diagnostic solutions throughout the world. By operating on extremely small sample volumes (nanoliters to microliters), paper microfluidic diagnostics can be very rapid. Paper microfluidics often integrate the readout into the paper itself (e.g., as a color change), eliminating the need for equipment like microscopes or sensors and makes paper microfluidics essentially equipment-free; in fact, recent work has shown that cellular phone cameras can provide effective readout capabilities for paper microfluidics.

1.3 Other Microfluidic Technologies

The vast majority of papers published on design automation for microfluidics over the past 15-20 years have targeted two specific microfluidic technologies: electrowetting on dielectric (EWoD – often called "Digital Microfluidics") [83] and channel-based microfluidics featuring integrated microvalves which are controlled via external solenoid valves [103].

Fig. 5.1 depicts fluid transport and mixing in these three technologies. In paper microfluidics (Fig. 5.1a) liquid expands in a radial pattern from the application point, in accordance with the theory of capillary action; the underlying physics is no different than using a commodity paper towel to mop up a fluid spill. Transport and routing of fluids within the paper is handled through various printed hydrophobic barriers, such as wax-based inks [11, 88, 22, 84]. The wax barrier impedes radial flow, but there is no external source beyond the substrate itself which pumps the fluid. This is significantly different than either the actuation mechanisms employed in other popular microfluidic technologies or the transport of electrical current in semiconductors, as the forces that are applied to fluids (or electrical currents) in the aforementioned technologies are inherently directional. In electrowetting microfluidics (Fig. 5.1b), the hydrophobic surface coupled with the pattern of electrodes that are actuated by an externally supplied voltage controls the direction of wetting (transport). In channel-based microfluidics (Fig. 5.1c), an external syringe pump creates a force which becomes directional due to channel geometry; the same is true of peristaltic pumping, which is internal to the chip, but is controlled by external solenoid valves.

The geometry of a paper microfluidic device determines the volume of the liquids and reagents that are required to successfully complete an assay. The time required for fluid to travel through a (portion of) the substrate can be constrained by both upper and lower bounds, depending on the assay: the upper bound may be due to evaporation, the rate of chemical interactions, and ultimately the amount of time that a person may be willing to wait for results or possible sample spoilage, while the lower bound is typically determined by the minimum time for chemical processes to complete. Further, the materials used in the device (substrates, inks, fluids, reagents, etc.) need to be limited to avoid waste in order to maintain low cost while still delivering efficient, effective, and accurate results.

1.4 Passive Flow Devices vs. Traditional Circuits

The microfluidic field is experiencing a similar design trajectory in general and paper-based microfluidics is also beginning to approach the same boundaries as silicon-based counterparts but with additional constraints unique to the field. In principle, automated design of paper microfluidic devices – and the individual components that are used to construct them – takes inspiration from semiconductor design automation: paper microfluidics feature components (like standard cells or IP blocks) connected by fluid transport channels (similar to wires). Although there are many parallels between traditional circuit design and the proposed passive-flow technologies, the fundamental difficulties stem from the differences between electricity and fluids. When working with electricity, our design concerns stem from resistance, voltage, amperage, and heat. As long as we complete a circuit, the electrons will get there eventually.

When dealing with fluids, there are similar concerns when it comes to resistance to flow, capacity of flow and rate of flow, however, paper microfluidic devices are subject to limitations not found in silicon. Paper microfluidic physical design must account for the underlying physics of passive fluid transport (e.g., wicking)[42, 25], the physical properties of the paper substrate[18], reagent and test sample requirements, timing boundaries for both sufficient reaction time and test duration. The volume of liquids and reagents necessary for the assay must be present in sufficient amounts to assure successful completion. Additional concerns related to fluid dynamics include: gravitational effects, fluid surface tension, and substrate resistance.

Further complicating placement and routing, locations and distances are not necessarily discrete as successful device and layout generation may need fractional modification, therefore we have a continuous range of viable locations. The time required for fluid to travel through a (portion of) the substrate can be constrained by both upper and lower bounds, depending on the assay: the upper bound may be due to evaporation, the rate of chemical interactions, and ultimately the amount of time that a person may be willing to wait for results, while the lower bound is typically determined by the minimum time for chemical processes to complete. The materials used in the device (substrates, inks, fluids, reagents, etc.) may also need to be limited to avoid waste in order to maintain low cost while still delivering efficient, effective, and accurate results.

Any researcher or developer may have to take into account any number of the variables when developing new devices and device design automation must also account for non-discrete and non-polygonal geometries as components can be located anywhere within the device – unlike traditional circuit placement which is restricted by grid-oriented standard cells – which complicates the validation of placement legality. Lack of standardization suggests that design constraints akin to standard cells for paper microfluidic devices are unlikely to emerge in the foreseeable future. Thus, straightforward adaptations of existing physical design algorithms are inappropriate for paper microfluidics, and design tools and methodologies to address these challenges are needed.

1.5 Current Paper Microfluidic Design Practices

All paper microfluidic devices share certain advantages over traditional laboratory-based analyses, including low cost, ease of use, and little or no additional equipment needed [42, 65, 22]. However, the process of developing new paper microfluidic devices requires actually finding a design that works as expected. And once a functional device is found for one application, it may not work as intended for a different application, so the design process, including optimization, must be repeated again. While there are some mathematical models available for predicting the flow of fluids through paper [107, 63, 18], usually several rounds of device design and testing are necessary before a functional device design is found.

Despite the aforementioned advantages, designing, testing, and validating paper microfluidic diagnostics remains a significant hurdle, as the task is presently done by hand using software such as AutoCAD® or Adobe Illustrator®. Each design is therefore specific to the particular conditions and parameters under test as well as limiting the complexity of designs and potential biochemical reactions. During the development phase, the designer must then create multiple device variations to evaluate design performance and assay accuracy to ensure that the device performs as intended within varying environmental conditions. A researcher wishing to test variations in channel count or reagent combinations would necessarily have to manually re-draw each variant before even beginning to test any particular design for the desired assay environment. Under the current paradigm, these design variations must therefore be re-drawn for each application by hand. Such a process involves error-prone and intensive labor as well as requiring greater time and resource demands for the designer – instead of performing research into a paper microfluidic device.

Even though paper microfluidic devices are one-time use and disposable, during the development phase, researchers must create multiple variations to compare design performance and accuracy, and possibly to adapt the device for use in varying environmental conditions. Under the current design paradigm, the researcher nneds to hand-design each variation, essentially a “hard-coded” ASPMD (application-specific paper microfluidic device; analogous to an ASIC), which is time-consuming, labor-intensive and prone to inaccuracy. These issues also limit the potential complexity of the biochemical assays (step-by-step chemical reactions) that a paper microfluidic device can realistically be designed to perform.

As a specific example of the challenges facing designers, consider the paper microfluidic multiplexer [72], versions of which have been designed that contain two, four, six, or eight user-selected channels. Now suppose that a designer requires a paper microfluidic multiplexer with a different number of channels, or a different arrangement of the channels, or countless other possible modifications to the design. Currently, this would require the designer to manually edit one of the existing designs, or draw a new design complete from scratch, using a computer drawing program, prior to fabricating and testing a new device [66]. This process could iterate numerous times before a satisfactory new device design is found.

As the research into PMD’s develop, the increasing complexity and array of potential applications that the topic provides is driving the need to produce new tools and methodologies to handle the increasing sophistication of devices and further enable robust biological diagnostics and - potentially - new therapeutic applications. Increased complexity comes with the increased challenge of accounting for and incorporating the abilities and specifications that would provide the potential capabilities. The exponential growth of processor design very early on necessitated the development of design automation software that could generate and validate processor designs for fabrication.

Chapter 2

Related Works

To the best of our knowledge, there has been no prior work on design automation for paper microfluidic devices based on passive flow substrates; many individual devices, designed manually, have been reported [65, 68]. There has been some prior work on physical design and droplet routing [106], control pin optimization [104] and test [105] for paper-based electrowetting devices patterned using conductive ink [2, 56]; these devices transport liquid through the application of high-voltage electrostatic forces, as opposed to passive capillary flow. In principle, our framework could be used to print the conductive ink patterns on the paper. In the future, we may try to use our framework to build paper devices that integrate electrowetting circuitry with passive flow substrates.

Some work has been done using computers to automate the generation of microfluidic device designs; however, most of these approaches are limited to normal channel-based or "continuous flow" microfluidics, not paper microfluidics. For example, ten years ago, an AutoCAD plugin was designed to help optimize the generation of the control layer for valve-based devices fabricated using multi-layer soft lithography [4]; however, it requires the flow layer to be designed manually and does not offer design analysis or validation capabilities. More recently, a number of algorithmic approaches have been put forward which adapt principles originally established for semiconductor/silicon design automation, in which components are defined on a traditional grid [44, 79, 76, 16, 17]; however, they explicitly assume that fluid is contained in pressurized channels, which provide far more reliable and longer travel distances than can be realized using paper microfluidics. More directly related to this work is an open source Java program that aims to simplify paper microfluidic device design [19]. The application provides a graphical user interface from a set of basic geometric primitives, such as rectangles and circles, which lack behavioral descriptions. Thus, the end user is required to predetermine the desired and expected behavior(s) of each shape. To its credit, the application provides a scripting language to assist with design automation tasks for users with a programming background; however, it does not automate analysis or correctness checking for the paper microfluidic devices that it generates.

Most work recently published on design automation for microfluidics has focused on either paper-based electrowetting or channel-based devices. Chips using electrowetting [83] have fluid traveling on a hydrophobic surface that is routed via electrical signals. Several works [106, 104, 105] have investigated using a paper form factor whereby the signal circuitry is printed onto the paper, but still utilizes an external power source to actuate droplet movement, and I/O is presumably performed by pipetting, rendering the devices difficult to use. On channel-based devices [103], routes are fabricated in a solid material and fluid travels under external pressure and regulated by microvalves. These approaches can more appropriately utilize silicon strategies and methods due to adhering to a standard cell and grid approach and therefore do need a collision detection methodology or multi-layer considerations.

The use of indexing trees is well researched in the computer graphics field [57]. The *oriented bounding box* (OBB) tree is analogous in approach [34] but is limited in dimensionality. Several other variants [117, 67] employ a partitioning schema to reduce the zones for testing; in our case, extending the R-Tree guarantees minimal bounding containers in multiple dimensions serving a similar purpose. The *bounded deformation* (BD) tree [50] and general form sphere trees [82] are also applicable to 3-D models but lack the multi-layer generalization of the ND-Tree, as reported here.

Multi-dimensional work has been extended to 4 dimensions of spacetime collision detection [10] using binary tress and division of spacetime in a divide-and-conquer approach using basic bounds. Other approaches expanded to using polyhedra [15] and streamlined to using using sphere trees [45, 46] as well to progressively tighter spheres to gain additional speedup. A more complex approach uses discretely oriented polytopes [55, 117] as the basis for bounds testing and is the closest approach similar to the one used in this paper but without the orientation and equal number of dimensions requirement. Other approaches involve inclusion of physical dynamics [61, 80, 82] but are geared toward computer animation and gaming applications.

Canonical techniques for standard cell and discrete routing inform application of non-standard, non-discrete microfluidic spaces. For placement, the Radar approach is based on simulated annealing [54] where optimal locations are search for through comparisons of local minima and maxima over the global function which the properties of Bézier curves provide. For routing, the use of probes was taken from [77, 41] for finding valid routes for fluid channel generation. Grimmer [35] and Amin [4] used similar approaches but were limited to standard cell approaches but able to use pressure and control to constrain the design space.

Part II

Design Automation

Chapter 3

Design Automation

3.1 Framework

To support the automated development of paper-based microfluidic devices (PAD), this chapter introduces a novel software framework that is intended to target both researchers and developers ultimately service the general population. Compared to existing lab-on-a-chip technologies, paper-based microfluidics differs in terms of substrate technologies and point-of-care usage across a wide variety environmental conditions. This chapter addresses the contexts in which the software can address these challenges and presents several initial case studies that demonstrate the capabilities of the framework to produce workable and usable paper microfluidic devices.

Using the framework, paper microfluidic device developers can specify, prototype, dynamically generate, and test their designs. The framework provides the capability to reliably reproduce designs streamlined for *in-situ* fabrication. Additionally, the framework integrates with tools to test and analyze each paper microfluidic design to enable automated

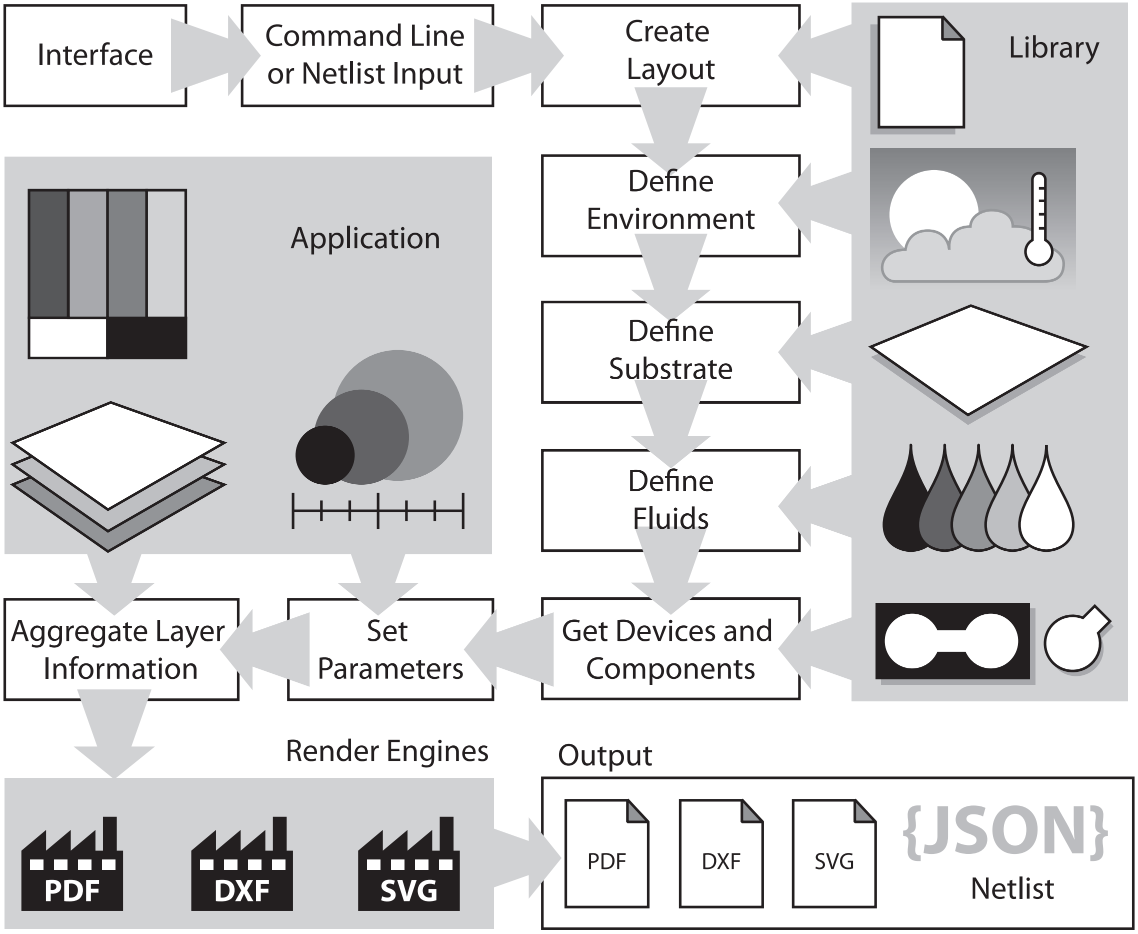


Figure 3.1: Framework overview which includes a library of paper microfluidic components, which can be rapidly assembled into netlists which form new devices. Once the netlist is assembled, the framework renders the device using established file formats (*PDF*, *DXF*, *SVG*).

or semi-automated design space exploration. For example, the framework could generate different variations of a design to account for the effects of environmental conditions, impact on physical substrates, and dynamic fluid conditions, providing the designer with a greater understanding of how these physical factors influence accuracy under test.

3.1.1 Segments, Paths, and Primitives

Segments, *paths*, and *primitives* refer to the elemental geometric shapes that can be printed onto a paper substrate. A segment is a Bezier curve, featuring source and sink coordinates that indicate a direction of drawing when rendered by the framework. The source and sink have handle coordinates that define the curve. If the handle coordinates are the same as the source and sink coordinates, then the curve degenerates to a line segment (Fig.3.2(a)) otherwise, they form a curve (Fig.3.2(b)).

Paths are constructed by concatenating multiple segments (Fig.3.2(c)). Primitives are closed paths, representing geometric shapes such as quadrilaterals, circles, ellipses, and polygons (Fig.3.2(d)). Union and intersection operations applied to Primitives can form complex shapes and Primitives with negative space (Fig.3.2(e)). The *Zero-sum Winding Rule* [48] can identify overlapping and negative spaces to determine the underlying paths that characterize the final shape.

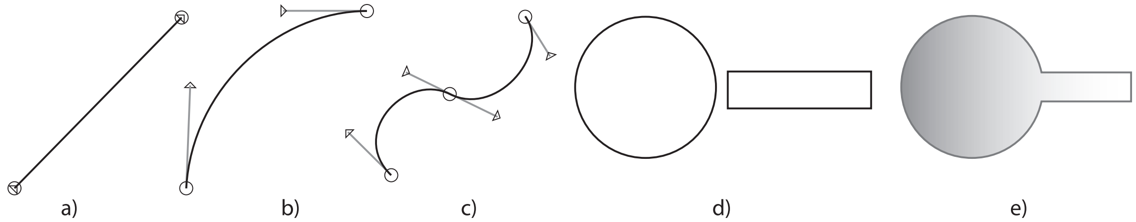


Figure 3.2: a) line segment, b) curve segment; c) a path constructed from two curve segments; d) a circle and a polygon primitive constructed from curve and straight segments; e) a component formed from multiple primitives.

3.1.2 Components and Devices

A *component* is a dynamically generated, re-usable object whose geometry is defined by one or more primitives, coupled with its functionality in terms of fluidic actions and abstract dynamics, such as mixing, transport, timing, etc.

A *device* consists of at least one or more components that encapsulate the desired actions and parameters needed to characterize biochemical behavior as described in the assay protocols. Individual components may be scaled or rotated as needed. Large devices may be specified hierarchically in terms of smaller devices, facilitating composition of multiple assays, either in sequence or in parallel.

3.1.3 Parameterization

The framework supports *parameterized components and devices*; for example, a straight fluid transport channel can be characterized in terms of its length and width. The device designer can then iterate over multiple versions, for example, to assess behavior of an assay under various loads, conditions, configurations, etc. This can be automated via dynamic drawing based on user-specified parameters. As the designer varies parameter values, the software adjusts and re-renders the component or device as needed.

3.1.4 Substrates and Substances

It is necessary to understand the properties of both the physical materials that the fluids that are expected to flow through it. The *substrate* abstracts the production of the output to individual pages (for printing) or other options (e.g., computer controlled-paper cutters). It encapsulates physical properties of materials, such as dimensions, margins, output area, expected flow rate, and fluid capacities. Attributes such as base flow rate for a given substrate can be used to calculate flow and dispersion times which can then determine expected execution time, timing variability, fluid consumption, and the probability of successful assay completion. Paper dimensions can be specified including margin size, along with mean flow rate for a fluid to travel within the substrate.

The *substance* encapsulates the physical characteristics of the various fluids, reagents, solutes, flow rates given a particular environment, etc., which are used to evaluate assay feasibility and accuracy, given a substrate.

3.1.5 Registries

Each device or component has a *registry*, which tracks and resolves the substrates onto which it will be printed. When two components or devices are merged, their registries are merged and reconciled to determine if the substrates and layers are compatible; if not, the device cannot be fabricated. If the device is feasible, a *layer registry* is built to determine which primitives, components, and devices will be printed on each layer. When appropriate, duplicates may be eliminated and components may be transferred from one layer to another, if needed. This reconciliation process produces a distilled list of substrates which the user can then verify for correctness.

3.1.6 Layouts

Paper dimensions can be specified including margin size, along with mean flow rate for a fluid to travel within the substrate. The *layout* represents the context in which a paper-based biochemical assay operate, including environmental information, substances and substrates, and the base units of measurement. The layout contains the netlist representation of each device, as well as its layer registry.

3.1.7 Scale and Color

Scale and *color* are application-level attributes that speak to the accuracy of components and may define the basic operating modes of a device. Scale refers to units of measurement (e.g., mm-scale devices), including conversion between units. The default internal unit of measurement is millimeters. Storing and maintaining a metric standard minimizes rounding error accumulation and reduces measurement error that can occur when multiple conversions between non-metric measurements cascade. The notion of color includes standard four to six color inks as well as non-ink materials such as wax or metallic ink for printing.

3.1.8 Environment

The *environment* provides measurements that may influence assay performance and accuracy, such as temperature, barometric pressure, and humidity; this allows designers to characterize the environmental conditions under which a device will properly operate. For example, should a warm temperature prolong flow rates, then various channels and other components should be shortened to maintain the expected runtime; in a humid environment, runtime may increase or cause incorrect mixing. Thus, it may be necessary to produce a general family of devices that are capable of executing one assay under different environmental conditions.

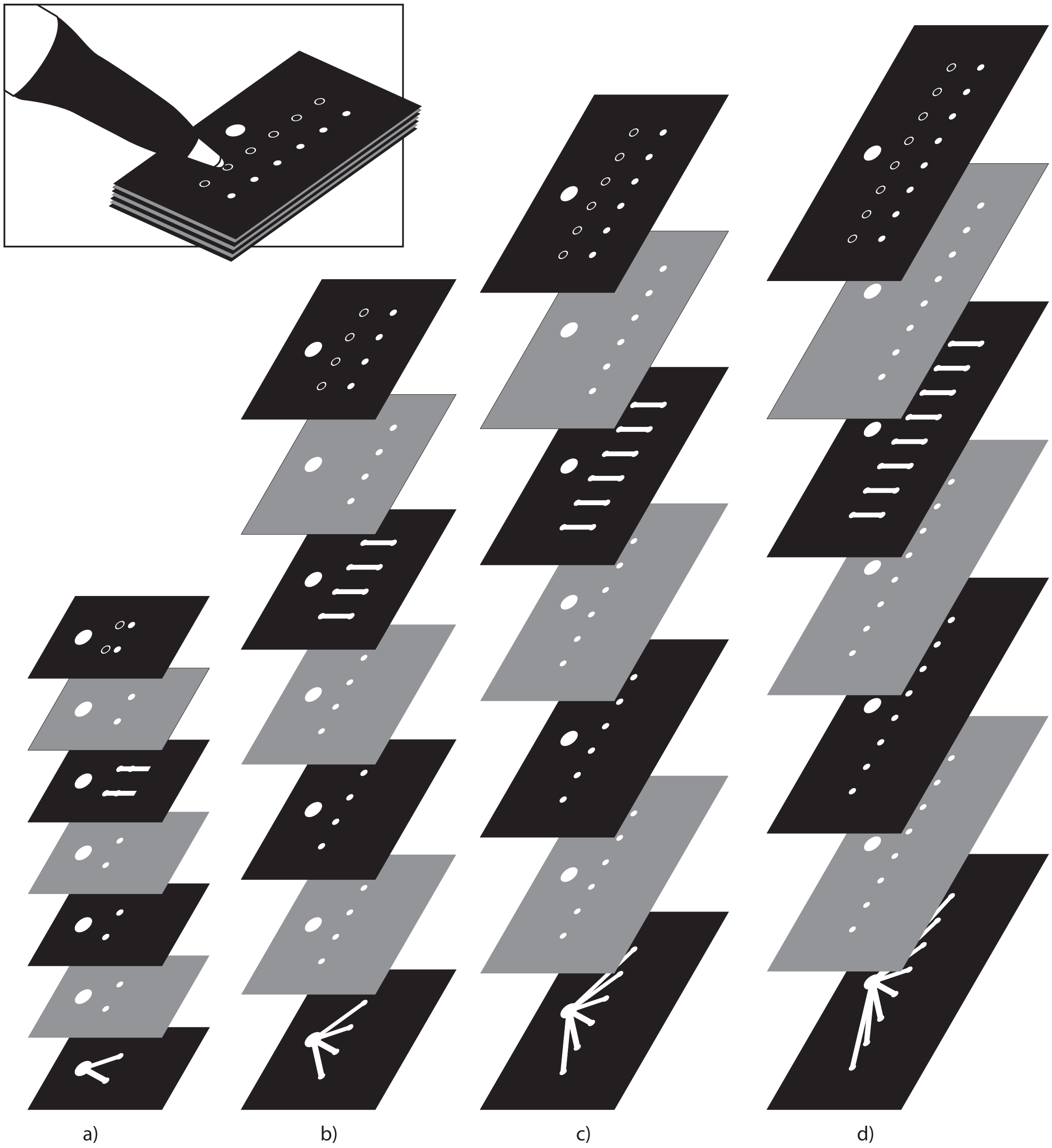


Figure 3.3: Parameterized fMux designs with a) two channels; b) four channels; c) six channels; and d) eight channels

3.1.9 Rendering Engines

The framework's initial rendering engine outputs the PDF format, which is ubiquitous, and supports vector graphics with high resolution output; in practice, the print or output device, not the PDF format, will limit the achievable resolution of our framework. PDF is platform independent, which simplifies distribution of device designs and enhances reproducibility.

The second rendering engine targets AutoCAD's DXF file format, which has been a stable of computer-aided design (CAD) for decades. The DXF is a platform-agnostic drawing technology that can produce highly accurate vector drawings. Due to the ubiquity of AutoCAD in computer-controlled machining (CNC) applications, the DXF format is compatible with cutting and pen-based drawing devices.

The last rendering engine targets W3C's Scalable Vector Graphics (SVG) format, an XML-based platform-agnostic file format. As SVG is widely used for web content, we expect to use this engine to distribute images primarily rendered for visual display and scientific dissemination.

3.2 Case Studies

We briefly summarize a set of successful case studies which demonstrate that our framework can be used to print a variety of useful parameterizable components and devices. These case studies validate the practical usability of the framework, along with its basic capabilities.

3.2.1 Fluidic Multiplexers

We used the framework to reproduce a *fluidic multiplexer (fMux)* [72] (Fig.3.3), which is constructed from multiple layers of paper and tape. The user "programs" the fMux by squeezing the layers together at pre-specified locations (buttons) on the top layer, connecting two passive flow substrates; the tape layer ensures that each button remains pressed (i.e., the two passive flow substrates remain in contact) after compression.

The user can program a K:1 fMux so that any subset of K input fluids will merge and mix (by passive diffusion) at the fMux output. Parameterization allows the design to generate fMuxes with any desired number of inputs; the user may also adjust the fMux dimensions, channel length, and well size. The framework then draws the fMux channels algorithmically using the rendering engine.

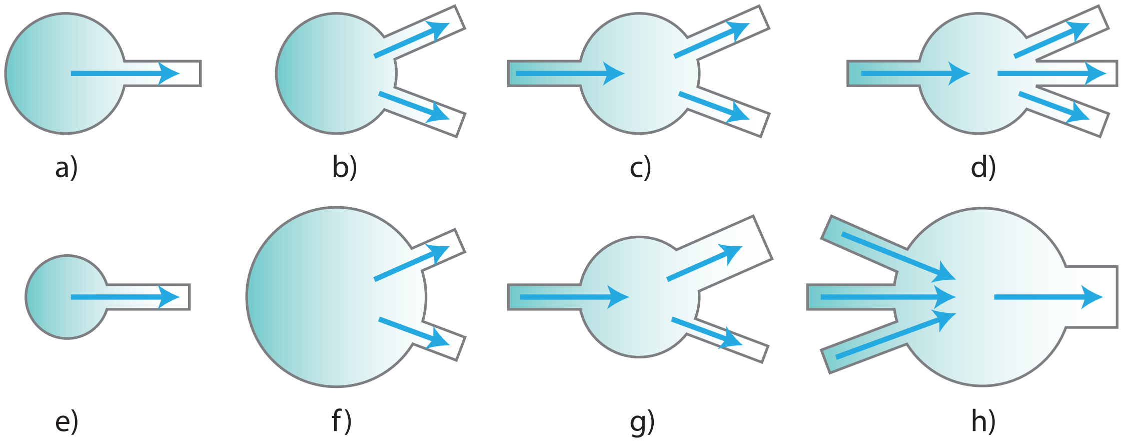


Figure 3.4: Well and channel components in 8 configurations: a) 1-out; b) 2-out; c) 1-in, 2-out; d) 1-in, 3-out; e) reduced well size with 1-out; f) increased well size with 2-out; g) 1-in, 2-out with channels of varying width; and h) increased well size with 3-in (narrow width), 1-out (wide width).

3.2.2 Channel and Well Components

Channels and wells (Fig.3.4) can be parameterized to change dimensions, capacities, orientations, and signal counts.

A *channel* is specified by its source and sink coordinates, along with its width; it may also be specified as a vector with a source coordinate, angle, length, and width. The framework can assess the impact of channel design parameters on issues such as assay execution time, reagent consumption, etc.

A *well* is a circular region that contains one or more fluids; the primary parameter of interest is its radius. Using the framework, we produced two calibration devices that capture performance data across varying environmental conditions. These devices can enhance reproducibility of scientific findings across a variety of external conditions which are often beyond the control of the practitioners.

3.2.3 Calibration Devices

The Raceway Calibration Device (Fig.3.5(a)) comprises a varying number of fixed-length lanes (channel segments) with varying width. The Raceway allows a researcher to compare fluid transport velocities over time, enabling characterization of dispersion rates. Running multiple "races" using a median benchmark fluid under fixed environmental conditions will enable a researcher to characterize the +/- margin of the substrate with a high degree of confidence.

The Bullseye Calibration Device (Fig.3.5(b)) allows for lanes to be specified at various angles in a radial distribution around a central source or sink. It includes a measurement scale that expands out from the central point in concentric marks giving appearance of a bullseye target. Paper substrates may exhibit a flow orientation as a result of manufacturing processes; unlike the Raceway, the Bullseye device allows for the calibration process to take the flow orientation into account, as it may influence accuracy and performance.

Once the calibration profiles are obtained, they may be incorporated into the layout of a device, especially when tuning the device for specific environmental conditions. We anticipate that this will increase accuracy, reproducibility, and efficiency in terms of time and material utilization, both in laboratory and point-of-care settings.

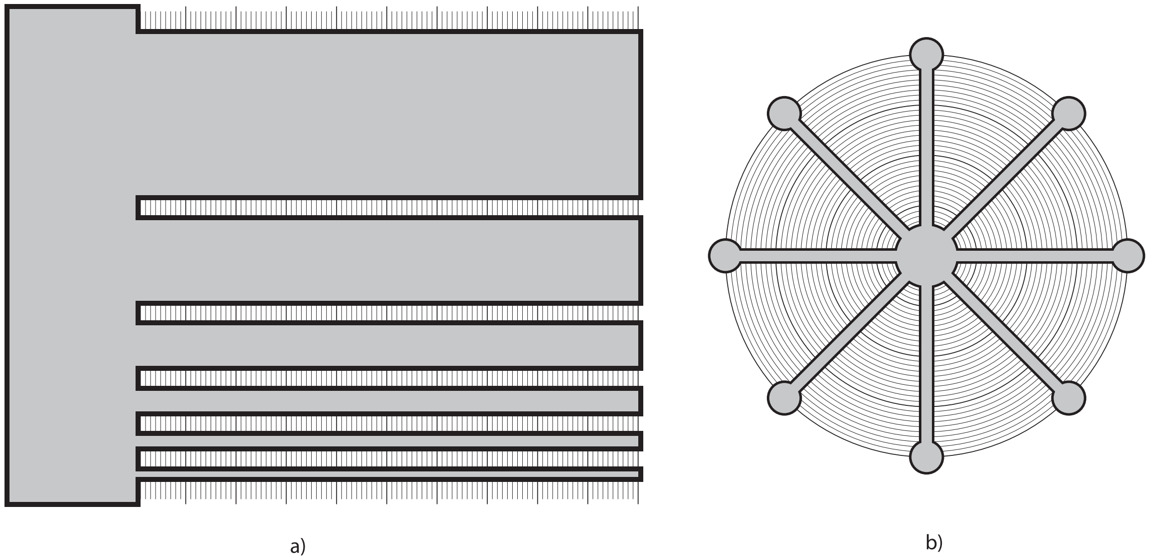


Figure 3.5: a) Raceway Calibration Device; b) Bullseye Calibration Device.

3.3 Experimental Results

This section summarizes the experimental setup, preliminary tests performed, observed difficulties, and solutions and processes that were develoepd to mitigate those difficulties. The reported experiments 1) determined a reliable width of the printed barrier walls to contain fluid transfer; and 2) assessed the reliability and replicability of experiments performed with devices produced by the software.

3.3.1 Experimental Setup

Microfluidic layouts generated by the framework for testing were printed using a Xerox ColorQube 8570DN wax-based ink printer connected via USB cable directly to a PC used for development. The printer driver's output resolution was set to 1200 dots-per-inch (dpi) with zero-scaling during output to maintain accurate rendering of files to substrates.

The wax printer delivers color solid wax "inks" to paper by activating an internal heating element, which melts the wax to a fluid state before depositing it on the surface material, where it rapidly cools before being ejected from the printer. Although the wax ink is hydrophobic, it resides on top of the paper, so fluids delivered to the substrate at this stage would penetrate beneath the ink, bypassing the desired barrier on the surface. Passing the printed page over a heating element re-melts the wax, which then flows into the substrate to create the desired hydrophobic barrier.

A 100-1000 μL range pipette was used for measurement and delivery of a solution of filtered water colored with a standard food colorant. LabNerd® filter paper cut to 200 mm × 123 mm served as the substrate.

3.3.2 Test Rig

In initial tests, the ends of the paper were folded over to create a rigid support (Fig.3.6, Left), which elevated the test area, but did not create a taut and level surface; consequently, when liquid was applied to the substrate, undesired flow would occur. Even with a 2 mm border to contain liquid flow to the desired regions, the uneven surface caused warping and uncontrolled and transport (Figure 3.7). The solution was to design and construct a test rig which isolates the substrates from other surfaces and maintains a level flow surface such to ensure that fluid is not under the influence of gravity while under test (Fig.3.6, Right). Results for subsequent experiments are reported using the test rig.

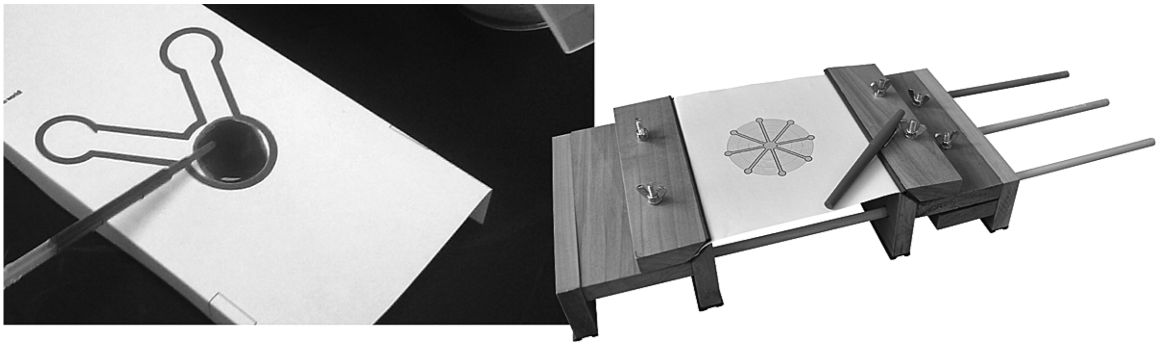


Figure 3.6: On the left, the first tests when creating wax-based channels for fluid flow. On the right, the test rig isolates the material and fluids from contact while under test.

3.3.3 Fluid Containment

The goal of this experiment was to determine a border width for reliable fluid containment for channels and wells under test. We ran several passes on the Raceway and Bullseye calibration devices using fluid volumes of 1000μL, 500μL, 300μL, and 250μL. For the Raceway device, 500μL of fluid caused a failure in the reservoir section, while 300μL reliably filled a Bullseye device with a 40mm radius. Subsequent experiments to test fluid containment were performed on the Bullseye device using 300μL of fluid with border widths of 0.5mm, 1.0mm, 1.5mm, and 2.0mm.

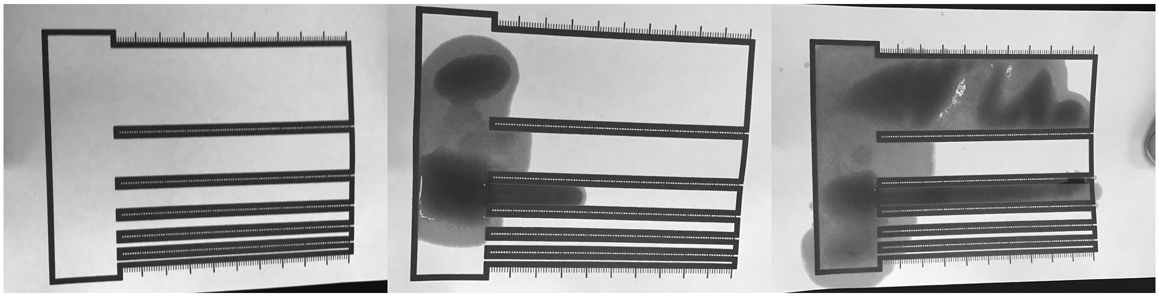


Figure 3.7: Simple experiment to test the Raceway Calibration device's containment efficacy. From left, @t=0 min, prior to dispensing fluid, @t=1 min, fluid contained by 2mm borders, @t=2 min, fluid flows beyond borders due to improper handling of substrate.

As shown in Figure 3.8, border widths of 0.5mm and 1.0mm failed to contain the fluid delivered to the source location in the center with fluid seeping through the channel barriers. The 1.5mm width performed better, but exhibited a failure at the 30mm mark (as indicated by the arrow in Figure 3.8), but otherwise contained most of the fluid. The 2mm width reliably contained the 300μL fluid, which was delivered to the source and then flowed to each of the eight sinks. In principle, this will allow for testing flow rates over varying device sizes; additional testing will attempt to determine the relationship (if any) of fluid volume to barrier size.

3.3.4 Flow Characterization and Replicability

Thirty 40mm-radius Bullseye devices were fabricated, two per sheet, and pipetted 200 μL of fluid to the source. For each device, the fastest and slowest times were measured

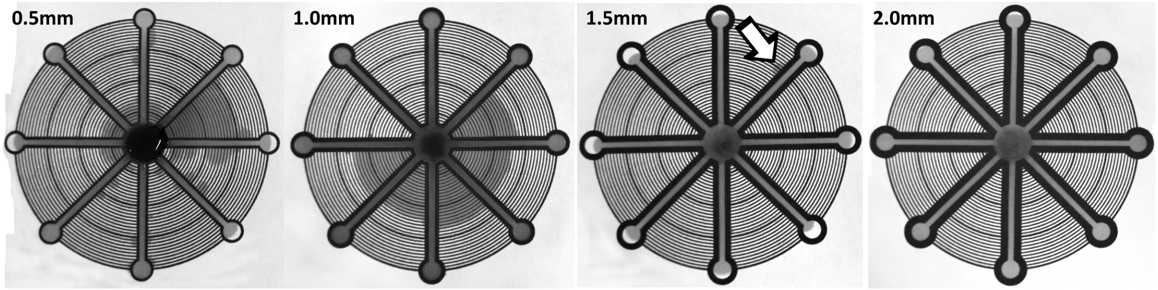


Figure 3.8: Fluid containment experiment to determine reliable border thickness for fluid containment. Note: photos had contrast adjusted to more clearly show fluid location)

at which fluid reached one of the eight 40mm sinks. The results (Table 3.1) suggest exceptionally high variability: fluid transport time ranged from $144 - 560$ s, with a median of 427 s and a standard deviation of 76.54 s. The primary cause for the variability turned out to be inconsistent pipetting, which, in turn, led to inconsistent fluid transport. Taken in its proper context, this variability is likely to manifest itself in real-world use cases, where the user of a paper-based diagnostic is a layperson, not a trained healthcare professional. Thus, there is likely to be similar high variability between different users of otherwise identical devices. Thus, experiments such as this can provide, at a bare minimum, upper and lower bounds on the time required to execute a biological assay on a paper microfluidic device; this type of information can and should be included in experimental protocols to ensure that the user has a realistic estimate of how long to wait before trying to interpret the results.

Table 3.1: Maximum and minimum fluid transport times recorded using the Bullseye device with eight $3mm$ sinks and $40mm$ long channels, having a $2mm$ border width, with 200μL fluid delivered to the source.

Trial	MIN	MAX	Trial	MIN	MAX
1	488 s	560 s	16	392 s	440 s
2	468 s	480 s	17	396 s	428 s
3	432 s	462 s	18	354 s	414 s
4	416 s	446 s	19	144 s	157 s
5	374 s	456 s	20	157 s	384 s
6	374 s	432 s	21	334 s	468 s
7	450 s	504 s	22	426 s	468 s
8	458 s	488 s	23	368 s	440 s
9	372 s	372 s	24	438 s	438 s
10	358 s	358 s	25	360 s	398 s
11	392 s	432 s	26	338 s	376 s
12	406 s	406 s	27	438 s	476 s
13	376 s	480 s	28	442 s	462 s
14	408 s	482 s	29	328 s	446 s
15	452 s	452 s	30	314 s	422 s

Chapter 4

Detecting Valid Placement across Multiple Dimensions

While numerous analogies between microfluidics and semiconductor technologies have been espoused, the physical differences between fluid dynamics and electrical current are significant which suggests that current trends in physical design for microfluidics must change course in order to be of practical use to designers. As a step in this direction, this chapter studies the problem of collision (overlap) detection during placement of components when designing multi-layer paper-based diagnostic devices. In this context, standard cells do not exist, components may have arbitrary geometric shapes, and dedicated layers for fluid transport may not be possible.

Determining valid placement of components in an effective and efficient way becomes a precursor to enable subsequent study of physical design algorithms in this domain. The solution put forward in this chapter is a generalization of the R-Tree, which is widely

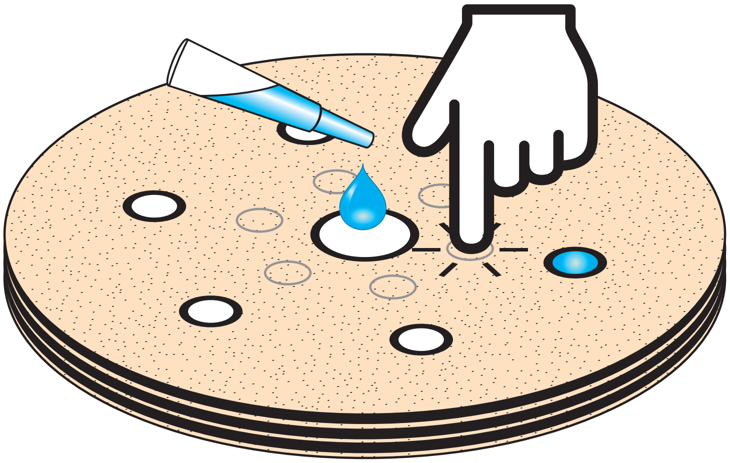


Figure 4.1: Multi-layer design version of a radial fluidic de-multiplexer (rMUX) based off a previously published fluidic multiplexer [72] as represented in our framework. The rMUX device accepts fluid delivered to the center well of the device and the user can indent the areas indicated by the grey circles to route fluid to the corresponding receptacles at runtime. Additional details are shown in in Figs. 4.2, 4.3, 4.6, and 4.7.

used in spatial databases, to address valid placement across multiple substrate layers and complex shapes. The new data structure called an ND-Tree can validate a placement across the entire substrate design space, down to the actual complex shapes of primitive components. Experiments demonstrate that the ND-Tree is as efficient as the R-Tree method to detect collisions between components yet provides additional encapsulation to span multiple substrate layers and fluid dynamics in a paper microfluidic device.

The increasing sophistication of paper-based microfluidic devices – such as the

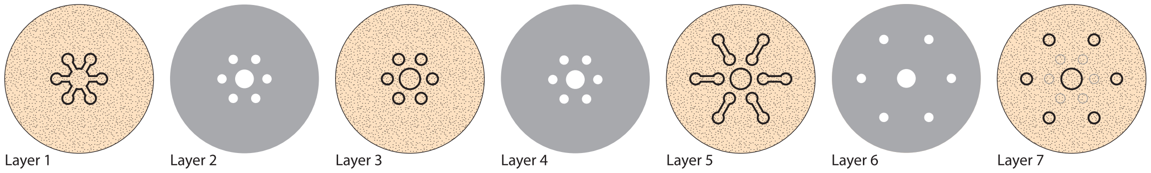


Figure 4.2: The individual layers of the rMUX showing the alternating layers of printed papers and cut hydrophobic barriers.

radial de-multiplexer (rMUX) featured in Fig. 4.1 – is driving the need to produce new tools and methodologies that enable more robust biological diagnostics and potential therapeutic applications. Multi-layer paper microfluidic devices introduce greater sophistication, which introduces the need for design automation techniques to place and route components, and to verify potential layouts during device development. The question that this paper addresses is how to efficiently detect collisions (overlaps) between two elements during placement, while accounting for the fact that the collision may occur on only a subset of the device layers. Our collision detection employs a data structure that we call the ***ND-Tree***, itself an extension of the R-Tree [5], a data structure widely used in information retrieval for spatial data access. The ND-Tree enables collision detection among shapes of arbitrary geometry in bounding containers that span multiple device layers and captures other bounding criteria such as fluid volume, fluid transport distance, and arbitrarily more dimensions.

After liquids are delivered to devices to the application point (Fig. 4.1), transport and routing of fluids is handled by using defined hydrophobic barriers, such as wax-based inks which can be printed directly to the paper [11, 88, 22, 84]. The various reservoirs and channel paths formed by these barriers are themselves constructed by individual components that when integrated into an application design creates the geometries that determine the volume of the liquids and reagents that are required to successfully complete an assay. Components such as the well and channels used to construct the rMUX (Fig. 4.3) assay, can occupy area and volume spanning multiple device layers (Fig. 4.2), and each of which have their own volume and timing parameters complicating automated construction and

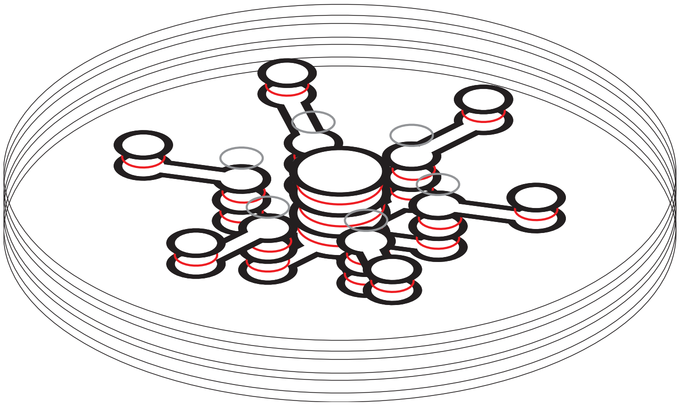


Figure 4.3: A six channel rMUX has seven layers to accommodate selection of fluid travel post-fabrication. The framework generates and configures six "via" Components that are placed according the number of channels specified during instantiation.

revision. All of these parameters must be taken into consideration when assessing potential locations of components in any assay being designed.

The diagram (Fig. 4.4) illustrates a case where collision detection based on bounding boxes alone can significantly affect the layout and the minimum sample requirements for a valid test. Using the bounding boxes produces an area of $1.92cm^2$ of waste area in the layout that would elongate the channels and require additional fluid volume and increased fluid travel. Given the $2mm$ channels, and additional $6mm$ vertical deflection and $16mm$ horizontal deflection would mean at least $((2mm \times 6mm)/100mm^2/cm^2) + ((2mm \times 16mm)/100mm^2/cm^2) = 0.12cc + 0.36cc = 0.48cc$ more fluid required in the layout. If a $10cc$ blood sample is to be tested, 5% of the sample is wasted due to the larger layout due to just this one wasteful placement. However collision detection of actual complex shapes involving curvilinear regions and boundaries requires far more intensive computation beyond simple area calculation.

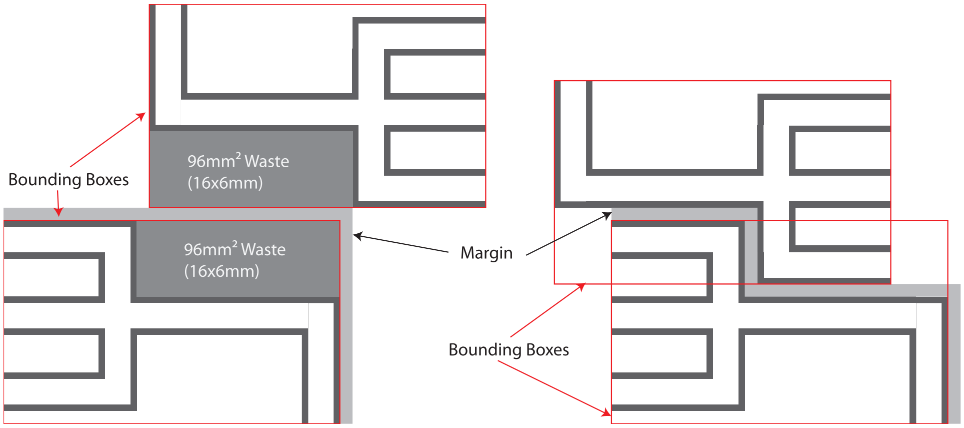


Figure 4.4: The left case demonstrates the problem of inefficient placement using bounding boxes only and the right shows placement according to actual shape with the dark grey indicating the space savings. The margin represents spacing that may need to be maintained due to ensure proper fluid containment.

Additionally, accurate placement evaluation necessitates comparison among all elements in the component to be place with all elements already placed. A naive approach would be to compare all elements against all elements. Consequently, multi-layer, multi-dimension collision detection is a precursor to physical design algorithm development for paper microfluidics. The use of a data structure like the R-Tree (Section 4.2), allows for comparing only those elements whose bounds intersect other bounds.

4.1 Design Automation Framework Review

The technical contributions presented in this chapter have been implemented the previously discussed for paper microfluidic design automation [86]. The framework includes a library of paper microfluidic ***components***, which are reusable objects that can be rapidly assembled into netlists to form new diagnostic device designs. A component definition may include functionality in terms of fluidic actions and abstract dynamics, such as mixing, transport, timing, etc. Each component $c = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ is physically defined by two or more geometric ***primitives***. A primitive ρ_i is constructed from one or more ***paths*** P_i, each of which using one or more ***Bézier curves***. Bézier curves are parametric arcs defined with start and end points and "handle" points that constrain the curve.

Within the framework, a ***device*** $D = \langle c_1, c_2, \ldots, c_n \rangle$ consists of at least one component, and encapsulates the desired actions and parameters needed to characterize its behavior. Each component is required to reside in exactly one device. Within that device, a component may connect to other components. Components may also connect to other components that reside within other devices. To create a device, individual components may be scaled or rotated as needed. Large devices may be specified hierarchically in terms of smaller devices, facilitating the concatenation of multiple assays in sequence or in parallel.

Devices reside on one or more ***layers***, each of which corresponds to a particular ***page***. Pages may contain multiple layers but are effectively special layers that also encapsulate what type of material or substrate it consists of, and its physical properties such as dimensions, porosity, density, etc. The terms "layers" and "pages" are used interchangeably as the collision detection method used in this paper is not concerned with the distinction.

A ***layout*** $L = \langle D_1, D_2, ...D_n \rangle$ contains all devices residing on one or more pages and also defines environmental variables such as temperature, humidity, and other variables as defined by the end user. Once a layout is completed, the framework renders the device using established file formats, such as PDF, DXF, and SVG.

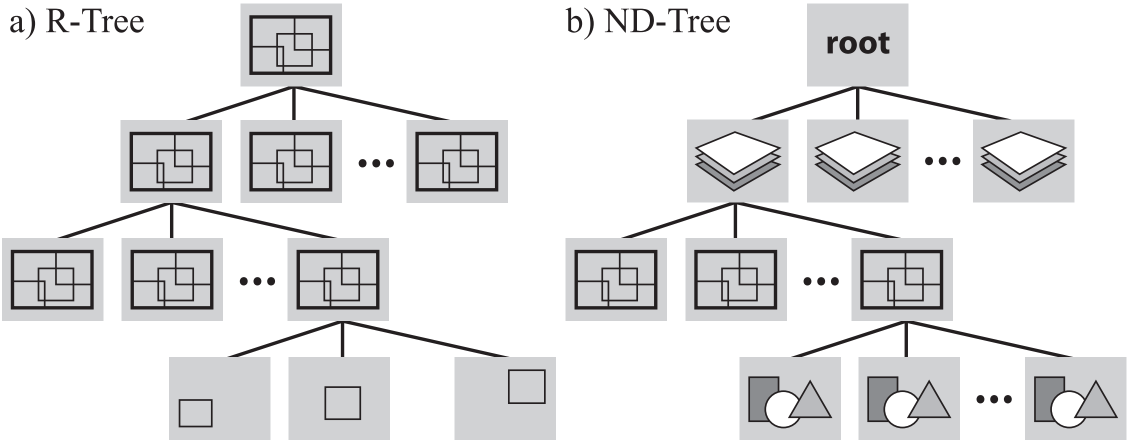


Figure 4.5: a) A tree contains n nodes, m children per node, and d dimensions per node. The R-Tree [38] represents physical space as minimum bounding rectangles (MBRs) that hierarchally contain other MBRs at the root and interior nodes, while leaves represent 2D points. R-Tree construction allocates MBRs that cluster nearby data points and, hierarchically, nearby MBRs. Nearest neighbor query comparisons traverse the R-Tree from the root toward the leaves, but may terminate early if the queried points are distant from one another. b) In a paper microfluidics application using an n-dimensional ND-Tree, the root establishes the overall layout, with the first set of nodes being grouped according to the various substrate ID's in which the primitives reside. Subsequent nodes contain bounding comparisons in the form of n-dimensional containers. Collision tests among the Primitives occur within the leaves.

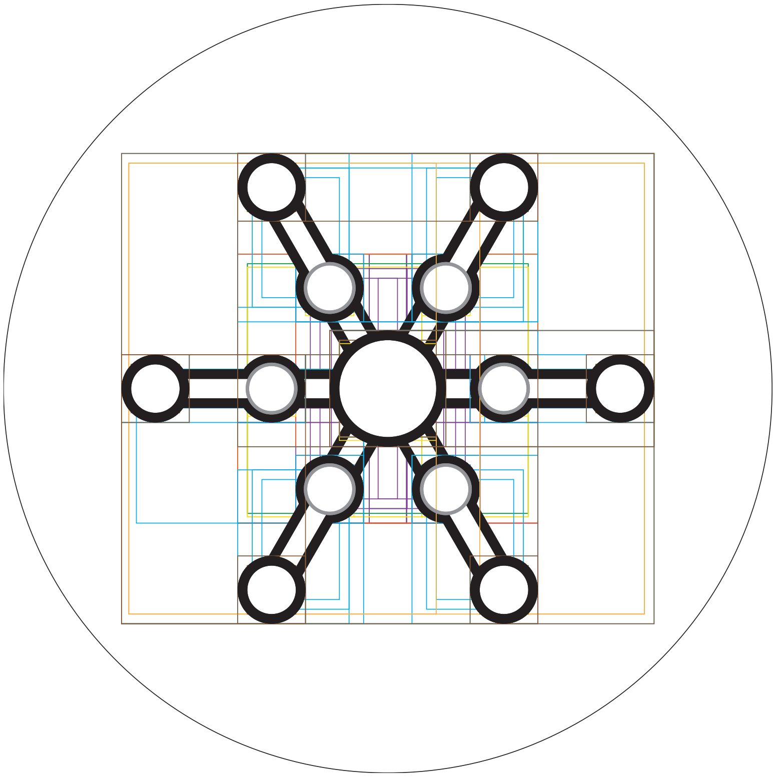


Figure 4.6: The rMUX with all primitives overlaid with its corresponding bounding containers.

4.2 The Naive Approach and The R-Tree

The ***Naive*** approach to detecting a collision between elements without any additional data structures would require pairwise enumeration; moreover, collisions are not transitive. However, logically grouping elements near to one other enables coarse-grained collision testing on the granularity of groups.

The ***R-Tree*** [38] (Fig. 4.5a) was originally introduced for locating regional points in a 2D spatial database stored in the leaves of the tree. Above the leaves, each node represents a ***minimum bounding rectangle (MBR)*** that contains the m nearest data points; the m children of the node are the leaves that represent those data points. Further up the tree, each node represents an MBR that contains the m nearest MBRs in an analogous manner. At the root, a single MBR contains all intermediate MBRs, which through recursion, ensures that the root MBR contains all data points. The *locality property* by which each MBR contains m *nearby* MBRs (as opposed to m arbitrarily chosen MBRs) is highly advantageous in nearest-neighbor database queries. The R-Tree provides an efficient nearest-neighbor indexing method that allows data points to be queried in $O(log_m n)$ time.

Fig. 4.6 illustrates the rMUX, shown with the associated bounding containers in a single-layer approach, using only the x- and y-dimensions in a straightforward R-Tree implementation. Fig. 4.7 shows the same device in the ND-Tree revealing the multi-layer aspect of the device. The R-Tree can capture each layer individually, and by using multiple R-trees the entire device can be indexed, however, the conceptual linkages of elements between layers, such as fluid channels connecting reservoirs, are lost. Additionally, the R-Tree must compare containers among all dimensions and does not encompass maximal bounding nor prioritize the factors being compared. The ND-Tree addresses these limitations while encapsulating all dimensions within a single unified data structure.

4.3 The ND-Tree

The proposed n-dimensional bounding tree (Fig. 4.5b), which we call an ***ND-Tree***, generalizes the structure of the R-Tree by allowing for any user-defined set of dimensions. These dimensions can be arranged in a sequence to enable determination of the amount of overlap between objects to support efficient bounds detection in a multi-layer, multi-dimensional space. One possibility is to optimize the sequence to minimize fluid vol-

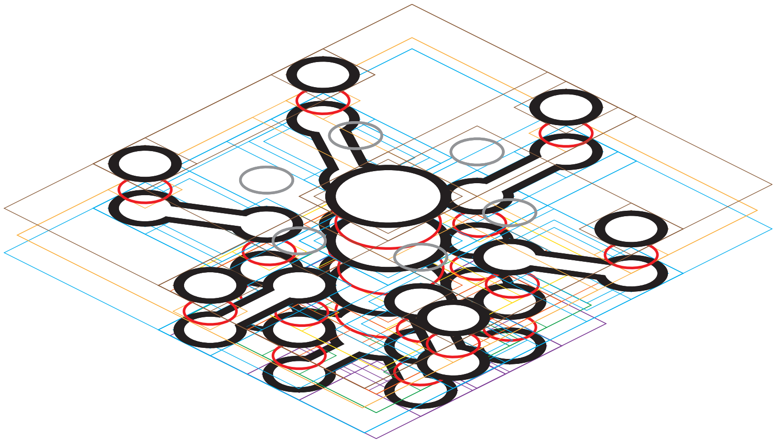


Figure 4.7: The primitives of the rMUX and the associated bounding containers in its corresponding ND-Tree. Each layer is shown in a separate color and demonstrates that even simple devices can be complex when determining their bounds.

ume by seeking to minimize the planar fluid area. Another option might be to ensure that the fluid spends a minimum amount of time traveling through a particular component to meet an timing requirement specified by the assay designer. A third possibility is to optimize the sequence for optimal placement and verification during location exploration.

The ND-Tree follows the same overall structure as the R-Tree with a root node which may have zero to m child nodes, n leaf nodes where n is the number of primitives placed in the ND-Tree, and internal nodes that track the bounds for each node and each of its m children. Fig. 4.7 depicts an ND-Tree that encapsulates all elements in on each layer of an rMUX device.

4.3.1 Nodes in an ND-Tree

An ***ND-Tree node*** (Fig. 4.8) consists of: a memory address to a node's parent if present, an m-sized array of memory addresses to the children of the node, a memory address to a primitive if present, and a minimum bounding container (MBC) that handles bounding dimensions in the internal nodes and primitives. There are five node categories: 1) a ***root***, 2) a ***graft***, 3) a ***branch***, 4) a ***leaf***, and 5) a ***fruit***. The root node has no parent and has exclusively graft nodes as children. The root is the only node type which has no limits on the number of children that it may have. A graft node serves as the funnel point for logical grouping of elements, e.g. by page. Graft nodes may contain any number of other graft nodes as children or branches to accommodate hierarchical grouping of nodes. A branch node has a parent node and a varying number of children ranging from m_{min} to m_{max} with a default maximum of four children. The child capacity is an adjustable parameter to aid in the balance of node distribution in the tree and to address performance optimization concerns. The allowance for a variable m also accommodates the varying nodes which can represent both a component constructed from primitives, and a complete paper microfluidic device constructed from those components. A leaf node will always have one parent node, but its children are exclusively fruit nodes. A fruit node stores primitives; all maximum-length paths in the tree originate at the root and terminate at a fruit node.

4.3.2 Encapsulating Dimensions

The angular bounds of an R-Tree can be generalized into a series of one dimensional bounds but in doing so, the concept of geometrical shape to define boundaries collapses beyond three dimensions. The ***minimum bounding container (MBC)*** supports additional dimensions that can represent desired characteristics beyond spatial constraints, for example, timing-related properties or fluid dynamic constraints. The MBC is defined as set of dimensions $\Delta = \langle \delta_1, \delta_2, ..., \delta_d \rangle$ that define the bounding conditions such as ensuring mini-

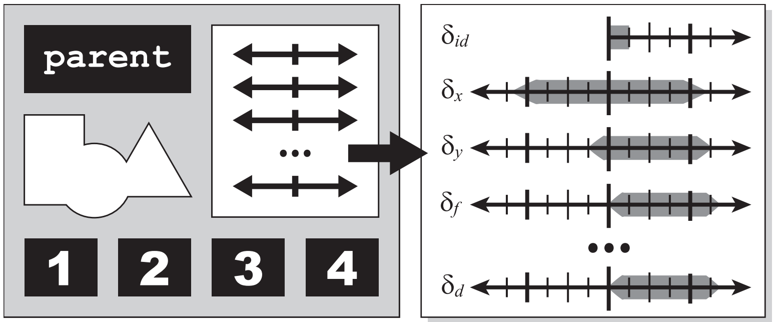


Figure 4.8: An ND-Tree node contains memory addresses for its parent node and all child nodes (black boxes). Child nodes are contained in an array to accommodate various configurations of and quantities of child nodes, depending on the state of the node, with 4 being the default maximum. If the node state happens to be a fruit, it will contain a pointer to the primitive (as indicated by the white shape) but will have no children. The bounding container is determined by the dimensions specified by the end user and are not limited to only x-, y-, and z-axes, but can feature any number additional dimensions that the ND-Tree can track. Here, values for layer ID, δ_x and δ_y bounds, as well as δ_f fluid distances are depicted, as well as any additional δ values.

mum fluid travel and volume constraints. Dimensions are defined and ordered by the user to ensure desired optimization parameters and to encourage early termination of a collision test, noting that there is *no collision* if there exists no overlap in at least one dimension (i.e., no layer violation). As an example, placing the page index as the first dimension to be tested ensures the only primitives and bounding nodes existing on the same page could possibly collide and thereby avoiding unnecessary testing.

Each dimension in the MBC also has a flag for determining the optimization direction, maximize, minimize, accrual, and neutral, which contributes to calculating optimal bounds. As previously noted, some assay parameters may have minimum timing constraints (e.g., a process must last for *at least t* time units), so optimization strategies that uniformly attempt to minimize execution time without accounting for these constraints may not generate a usable layout, even if all design rules are otherwise satisfied. The accrual mode sums each child's contribution to the dimension, e.g., to maintain a running to total of fluid volume required for a component. A page index does not have an optimization attribute, so it would be flagged as neutral.

The MBC (Fig. 4.8) of a node in the ND-Tree is constructed recursively as the dimension-wise union of the MBCs of its children. In the example, the MBC is tracking page ID, the x- and y- dimensions, and flow rate (as well as possible additional dimensions up to δ_d). When performing dimension testing, page ID would occur first; should a "collision" be detected, testing would proceed to x-, y-, and flow rate dimensions, until either no collision occurs or all dimensions have returned collisions.

4.3.3 Constructing an ND-Tree

The initial (empty) tree consists solely of a root node. Node insertion proceeds in a top-down manner where at each node in the tree, the insertion algorithm computes the MBC expansion for each child that would occur if that child's MBC was expanded to include the MBC of the node being currently inserted. To accommodate the generalization of dimensions and their according effects of minimizing and maximizing, node selection relies on the order of dimensions presented during construction to determine what is considered an "optimal" bounding container.

All the dimensions under consideration are approached as a Pareto set selection [43] method by which a grouping of dimensions is optimized together instead of individually. In an R-Tree, the expansion is determined as a function of area and/or volume of x, y, and z dimensions. The Pareto approach can weight the desired characteristics without the

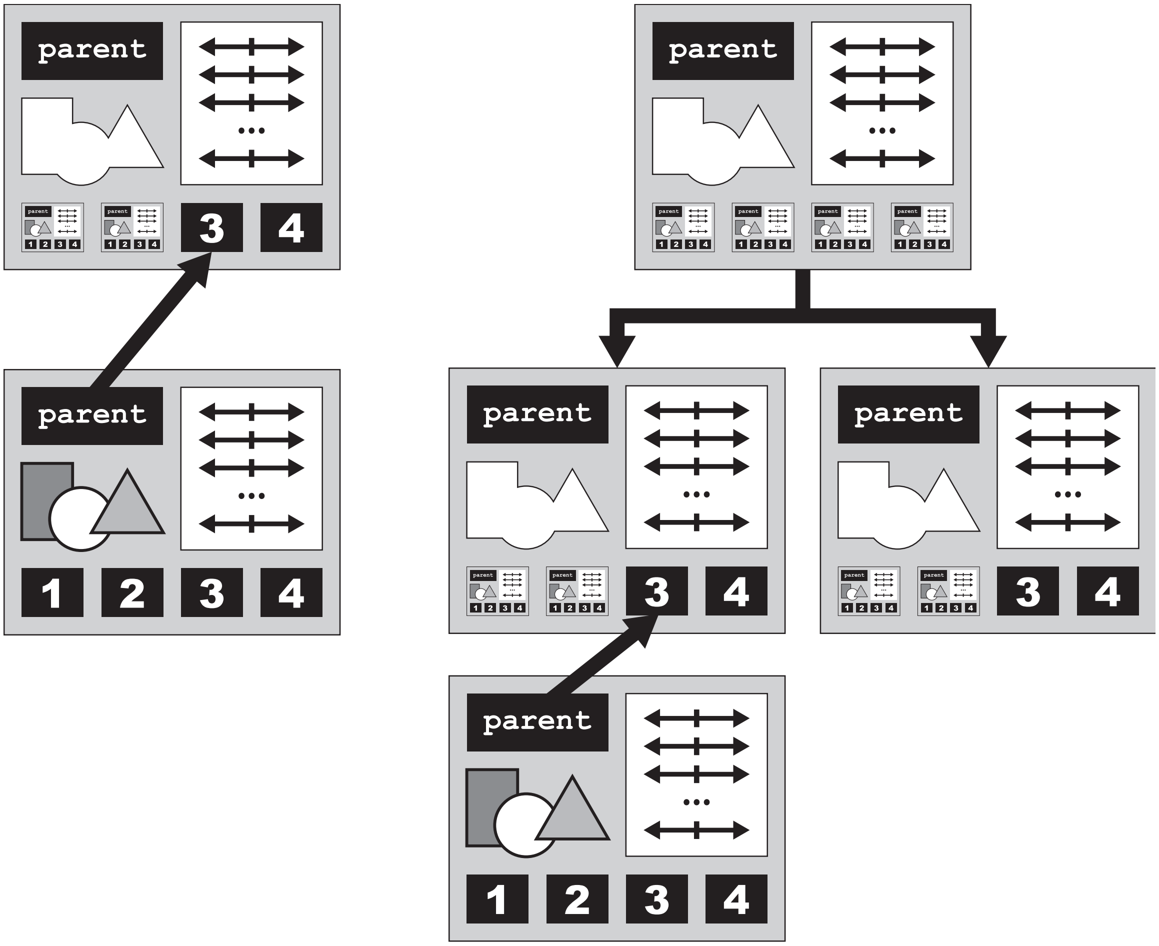


Figure 4.9: There are 2 cases to consider when inserting a new node into the tree depending on whether or not the node has room for more children. On the left, a new node with a primitive is to be added into the tree at the node above it. There are currently 2 free slots in this node and inserting the new node requires nothing more than selecting slot 3 and adjusting the bounds for the node. On the right, the node selected for inserting into is currently has no slots available so the node must be split and its children distributed between the now split node and the original node. Children are distributed to each node depending on the amount of bounds expansion. The new node is then placed in the node that also requires the least expansion.

difficulties of a mathematical function that would have to take into account non-physical dimensions that would have no meaning in a strict axial relationship.

The process is then recursively applied to the child whose MBC expansion is minimal, and repeats until a free slot is found at some level of the tree, or it reaches a leaf whose capacity is full (Fig. 4.9). In the latter case, the leaf is split into two new nodes, partitioning the MBCs between them. The MBCs of the two node may overlap. For each node, the objective is to minimize the weighted product of the dimensions of its MBC; the default weights for each dimension are 1, but the user can override them with different weights if they desire to do so. One-by-one, MBCs from the split node are processed. Each MBC is added to the new node whose MBC expansion is minimized. Additional constraints are imposed to ensure that both new nodes receive at least m_{min} MBCs. The new nodes are then re-inserted into the tree using the top-down approach described above.

4.3.4 Collision Detection between ND-Trees

The ND-TreeCollision function (Algorithm 1) takes as input two ND-Tree nodes to be compared, N_1 and N_2, as well as a boolean flag, β, to calculate overlap if desired, and returns either true or false indicating collision (Fig. 4.10 The algorithm is called, initially, by passing in the root nodes of each ND-Tree and where collisions are detected, the colliding nodes are recursively passed into the function. At each call, the base case, which occurs at the leaf level, is tested. If either of the two nodes is a branch node, the algorithm is called for each MBC in the source tree against each MBC in the test tree. The algorithm accounts for differing heights in trees by recursively calling the children of

Algorithm 1 ND-Tree Collision Detection

```
function ND-TreeCollision($N_1, N_2, \beta$)
    if $N_1$ is a $leaf$ and $N_2$ is a $leaf$ then
        return PrimitiveCollision($N_{1.\rho}, N_{2.\rho}, JP\beta$)
    else if $N_1$ is a $branch$ and $N_2$ is a $leaf$ then
        return ND-TreeCollision($N_1, N_{2.leaf}, \beta$)
    else if $N_1$ is a $leaf$ and $N_2$ is a $branch$ then
        return ND-TreeCollision($N_{1.leaf}, N_2, \beta$)
    else
        for each child $node$ $i$ in $N_1$ do
            for each child $node$ $j$ in $N_2$ do
                ND-TreeCollision($N_{1.i}, N_{2.j}, \beta$)
            end for
        end for
    end if
end function
```

the MBC in the taller tree with the leaf node of the shorter tree until the recursion bottoms out for both trees, and two leaves are being compared. If the algorithm reaches the leaf nodes, the more expensive PrimitiveCollision function is called.

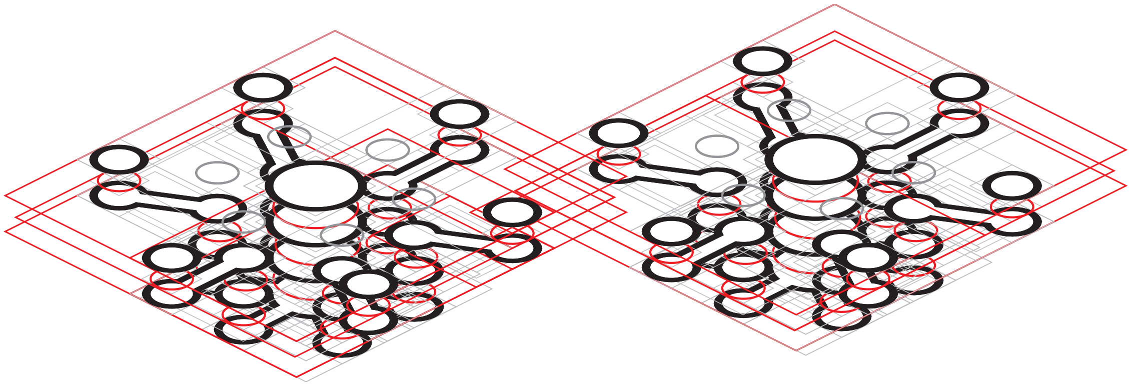


Figure 4.10: The collision of two ND-Trees using the bounding containers as shown in Fig. 4.7. Here, the red outlines show which containers are colliding among the all the elements in each component, noting that there is no actual collision between any elements in the components.

4.3.5 Detection of Primitive Collision

Once the collision detection algorithm (Algorithm 1) has reached the leaf level, the PrimitiveCollision algorithm selects the appropriate collision detection algorithm for primitive pair. Relatively simple primitives such as rectangles and circles can use basic geometric tests; in contrast, the computationally complex approach of searching for intersections of Bezier curves [115] is required when both primitives are curvilinear objects or basic geometric shapes that have been rotated or otherwise transformed.

As an example, Fig. 4.12 shows a case where two primitives P and Q are tested for a collision. Each segment in P is compared with each segment in Q, looking for intersection

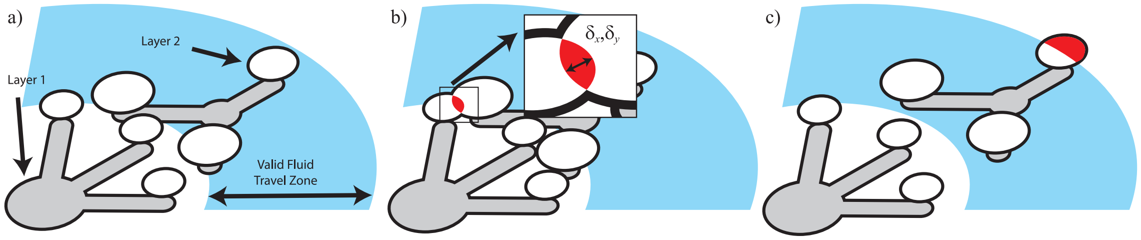


Figure 4.11: a) A simple example of collision detection and overlap calculation over multiple layers with a fluid travel limit. b) The calculate overlap of the 2 components existing on the same layer. The values δ_x and δ_y returned from the collision test will be the needed displace in those two dimensions to clear the device from collision. c) In this scenario, the collision test returns not the collision of elements in the device, but the crossing of the fluid travel limit barrier indicating that fluid will not make it to the end sink.

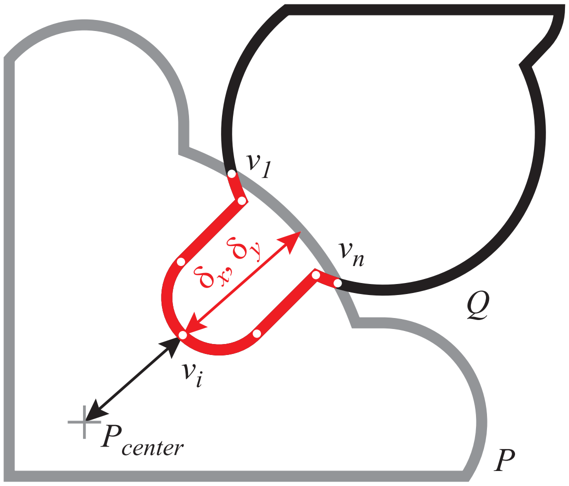


Figure 4.12: The collision of two primitives P and Q necessitate computation of intersection points between two or more Bezier curves. In this example (cropped to enhance detail), Q intersects P in two locations v_1 and v_n and a sub-curve of Q bounded by those two points lies within P. To determine the amount of overlap, each critical point on that sub-curve has its distance to the edge of P calculated by drawing a line from the center of P through the point v_i to intersection with the edge of P. The largest distance δ_x, δ_y between v_i and the edge intersection is returned as the amount of overlap of P and Q.

points between them. Bounding containers for segments are calculated during instantiation. The bounding container for the path is the union of the bounding containers of each of its segments. To determine if two segments intersect, each is recursively split into two shorter sub-segments until the size of each sub-segment's intersection bound is below threshold [52]. At that point, if the bounds determine that a collision has occurred, the original segments are determined to intersect and the location is registered as an intersection point and stored in an array in primitive P. If there is no overlap at any point, then the curves do not intersect.

4.3.6 Determining Displacement

The framework can optionally compute the amount and direction of overlap in each dimension which is necessary to displace elements to a "safe" candidate location in a layout. The user can leverage this feature to quantify the amount of movement needed to incrementally rectify a collision. Since a collision can occur in both the positive and negative directions, *tt* is therefore necessary to ensure that the amount of overlap detected is calculated across all layers and all dimensions, as shown in Fig. 4.11. This ensures that objects contained in the ND-Tree can be displaced by exactly that amount to potentially rectify a collision.

Overlap calculations occur within the ND-TreeCollision algorithm if the boolean flag β is set. As the calculations are performed the results for each dimension $(\delta_1, \delta_2, ..., \delta_d)$ are stored internally in the Δ tuple. The overlap is necessarily calculated from the bottom of the colliding primitives at the leaves, and is propagated upwards toward the roots. As

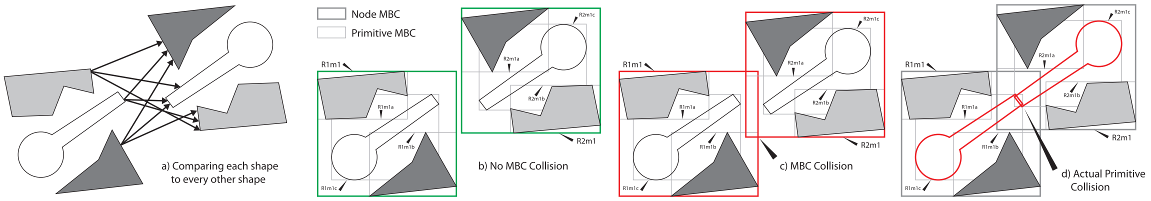


Figure 4.13: Collision detection cases: a) Without the use of MBC's b) No collision between two MBCss. c) MBC R1m1 collides with MBC R2m1 indicating that additional collision testing via ND-Tree recursion is needed. d) Collision at the primitive which will return a non-zero overlap.

shown in 4.12, the two colliding primitives P and Q have previously had their intersection points calculated. The WindingRule algorithm is then applied to these intersection points to identify the sub-curve of Q that is contained in P.

The WindingRule [47] determines if a point is contained in a shape by comparing whether or not the point is to the left or right of each segment in a primitive. If it is to the left, the winding number is set to 1 and if it is to the right it is set to 0. Once all segments have been checked, if the winding number is 1 the point is inside the primitive, otherwise it is outside. For each point that is identified as being contained in the primitive, a line is drawn from the center of the primitive through the point until it intersects with the edge of the primitive (Fig, 4.12). The distance between that intersection point and the initial point is calculated and the distance of point on the sub-curve that is farthest away from the edge is returned as the overlap amount.

The MinMaxDelta function is called to compare the returned value against the running value of the recursive calls and returns the amount of overlap that occurs between the trees and the direction in which the collisions occur relative to the point at which the collision detection was initiated. The calculated overlap contains both the minimum and maximum of the two overlaps for each $(\delta_1, \delta_2, ..., \delta_d)$ in tuple Δ. This provides the caller with enough information to clear the collision by relocating one of the two devices, components, or primitives corresponding to the ND-Trees (Fig. 4.11).

4.4 Complexity Analysis

This section analyzes the time complexity of three possible collision cases: 1) no collisions of any MBC's or Primitives between the two trees, 2) a complete collision of every element in both trees (such as alignment between the trees), and the common case of 3) partial collisions between the trees where there are collisions between two or more MBC's, two or more Primitives, or a combination of MBC's and Primitives. Let d be the number of dimensions, u be the number of substrates, p be the number of primitives, m be the number of children per node in the ND-Tree, and n be the number of MBCs.

4.4.1 Naive Method

The naive approach, which we use as a baseline, performs a pairwise comparison of all primitives in objects A and B. Without any indexing or additional data structures, the naive approach performs a pairwise collision between every element in one object to every element in another object – and in every dimension. This approach yields an overall time complexity of $\Theta(\Delta_d p^2)$ per layer.

4.4.2 R-Tree and ND-Tree Collision Methods

Both the R-Tree and ND-Tree employ a dominance test among all of their dimensions in order to determine collisions. As the ND-Tree employs an arbitrary number of dimensions, the order in which dimensions are tested can significantly impact performance. Examining "critical" dimensions early can maximize the likelihood of early exits in the common case where collisions do not occur. Additionally, logically grouping dimensions to specific grafts can "prune" paths in the tree while also maintaining their conceptual relationship. e.g. grouping by page or particular function, which may also provide further reductions in the number of comparisons.

No Collision

The first case as shown in Fig. 4.13b is a test of two components' or devices' ND-Trees in which the placements do not overlap on any layer. In this case, a minimum test of the bounding containers can ascertain than the respective MBCs do not overlap. This can be determined at the root of the tree, eliminating the need for a tree traversal. This process repeats for each layer, yielding an time complexity of $\Theta(\Delta_d)$ where d is the number of dimensions being tracked.

Total Collision

The worst case performance of the algorithm occurs when two or more identical components or devices are fully aligned across all dimensions. In this case, all MBCs intersect and induce a complete traversal of the ND-Tree, incurring a collision detection test at each leaf; collision detection tests are far more time-consuming than tree traversals.

This effectively degenerates to the naive method with the additional overhead incurred due to ND-Tree traversal.

Partially Overlapped Components

Let CD_{source} denote the set of components or devices that have already been placed, as indicated in Fig. 4.13b, and CD_{test} denote the set of components or devices currently being tested for collision. Initially, page ID in the MBC_{source} of CD_{source} for pages $s_1, s_2, ..., s_u$ are checked against $t_1, t_2, ..., t_u$ of MBC_{test} from CD_{test} to look for matches, which indicate that the primitives in CD_{source} may potentially collide as they reside in the same page. For each page s_i, the MBC for CD_{test} is compared against the MBC for CD_{source}. If MBC_{source} intersects with MBC_{test}, the ND-TreeCollision is recursively called for each intersecting MBC of CD_{source} and CD_{test} until either no further MBC's collide (i.e., the primitives contained within the original MBCs do not actually collide), or PrimitiveCollision returns the amount of overlap occurring in the collision (Fig. 4.13c).

Initial comparisons of CD_{source} page ID's $s_1, s_2, ..., s_u$ against CD_{test} page ID's $t_1, t_2, ..., t_u$ yield u^2 comparisons, given there is a fixed number of pages in a layout. Next the algorithm is called recursively for each $MBC_{test_j} \in CD_{test}$ that intersects with $MBC_{source_i} \in CD_{source}$, all the way to the leaf level at the maximum height of the ND-Tree in the worst case. Each MBC collision test requires two comparisons per dimension to test for inclusion. Testing for non-exclusion is more efficient than testing for inclusion, and has a time complexity of $\theta(d)$, where d is the number of dimensions being tracked. At the leaf, the WindingRule's time complexity based on number of segments a is multiplied by the number of primitives p being tested giving it a $O(a^2p)$. The overall time complexity is calculated as number of pages in $CD_{test} \times CD_{source}$ which is equal to u^2. Then $MBC_{source_i} \times MBC_{test_j}$ which is $i \times j$ nodes or n^2 nodes to a depth of $log_m n$. Lastly, the WindingRule time complexity, $O(a^2p)$, is added, yielding an overall time complexity of $O(u^2 + n^2 log_m n + ma^2p)$.

4.5 Experimental Results

We compared ND-Tree-based collision detection to the naive method and the R-Tree (Section 4.4). To enable a fair comparison, we modified to the R-Tree structure to have the same dimensionality as the ND-Tree: we added the z-axis, component area, and fluid volume within a component. This extended the R-Tree's MBR from a rectangle to a cube, and the data structure computes and compares volumes rather than areas. Page numbering does not conceptually contribute to the R-Tree's MBR, so we did not include it in volume-related calcusions. We implemented all algorithms in C++ v.11 and ran the comparisons using a custom test harness within a previously-published paper microfluidic design framework [86]. Experiments were performed on a desktop PC with an Intel Core i7 processor and 8 GB RAM running Ubuntu v18.0.1 Linux.

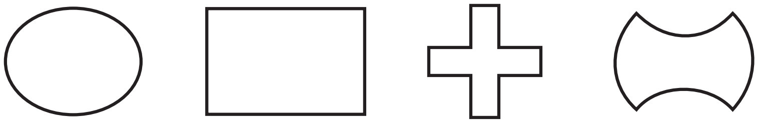


Figure 4.14: The 4 test shapes used in the random generation of devices that utilize the ND-Tree for collision testing.

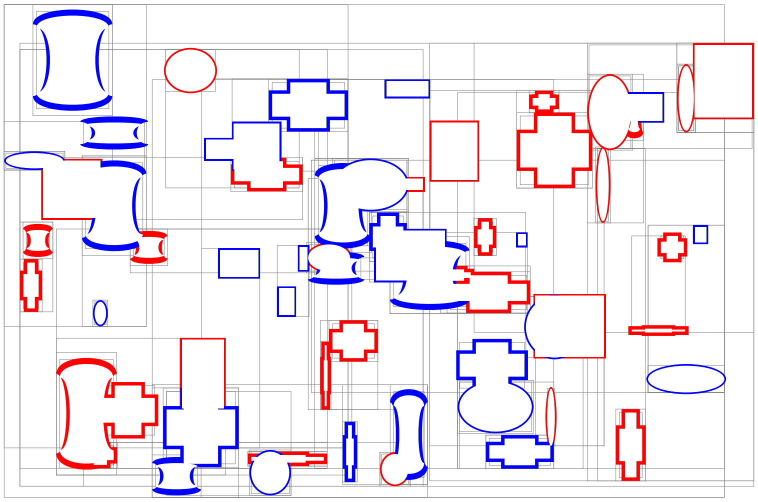


Figure 4.15: A representation of the experimental devices that were randomly generated for testing along with their associated ND-Trees. The source device is in blue and the test device is in red.

4.5.1 Methodology

For experimental evaluation, devices were constructed by randomly selecting from among four general primitive shapes (Fig. 4.14): ellipse/circle, rectangle/square, polygon, and Bézier-based curvilinear paths. These are basic shapes that could be expected to comprise components in a device and are meant to be representative of the classes of collisions being tested and therefore have the minimum shape complexity. Two test devices were constructed to evaluate collision detection for each experimental trial, similar to the example shown in Fig. 4.15. Each device has the same number of randomly chosen components, but with different configurations. Parameters detailing size and placement were randomized within a 40mm boundary to limit overall device area while maintaining a robust set of constructions for testing. One device served as the source and remained stationary while the its counterpart (the test device) was repeatedly placed at pre-defined locations.

A set pattern was programmed to ensure that collisions occurred by placing the test device in one of 10 positions. The initial placement positioned the center of the test device at the center of the source device. The test device was then displaced in eight directions, where the center of the test device aligned with the four corners of the source device as well as the mid-points between each pair of corners yielding nine potential positions to detect collisions. The tenth position placed the test device completely outside the bounds of the source device to ensure at least one collision-free run.

Each trial run consisting of 10 locations was performed 3 times, yielding 30 total collision tests per trial set. The number of components per device in each trial set were: $\{8, 16, 32, 64, 128\}$, yielding n^2 potential comparisons per collision test, i.e., a total of $\{8 \times 8, 16 \times 16, ...\}$ comparisons.

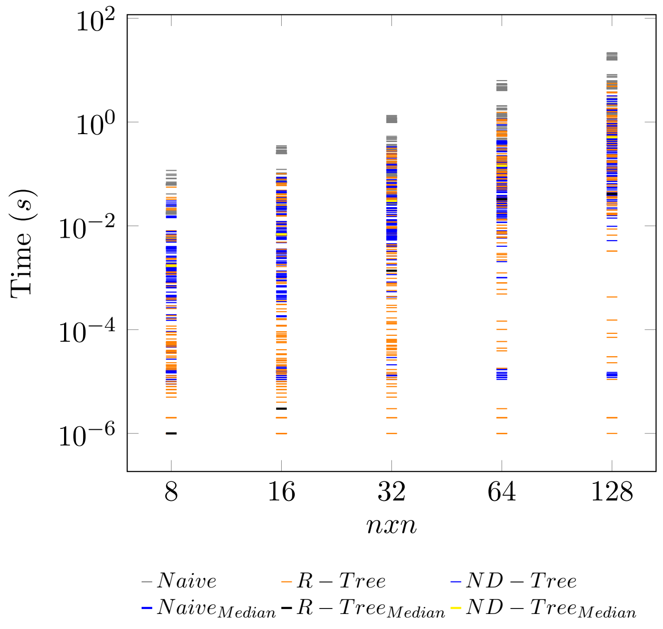


Figure 4.16: Scatter plot showing the runtime of each experiment, with the median highlighted. A logarithmic time scale was chosen to more clearly indicate the performance of each approach.

4.5.2 Results and Analysis

Fig. 4.16 reports the run times of each test, with the median run times overlaid. The median result characterizes the typical run time that would be incurred by an algorithm that calls collision detection. Fig. 4.16 includes some outliers in each case due to the variability of the randomly-generated primitives that make up the test devices. The ND-Tree has its greatest advantage when the devices increase in complexity. When this occurs, the initial overhead of the grafts is subsumed by the rest of the levels of the tree. Fig. 4.16 shows these outliers as part of the total run time, however, it does not distinguish them in terms of the complexity of the devices employed in the comparison.

The next step was to examine the best and worst-case performance of the naï, R-Tree, and ND-Tree approaches when collisions do and do not occur, and to quantify the additional performance load incurred when calculating device overlap.

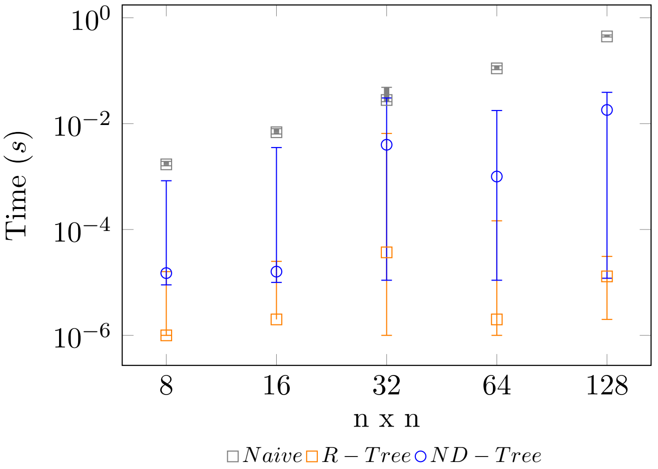


Figure 4.17: Runtimes for experiments in which no collision occurred among the devices located on a single page and the processing of the trees was limited to bounding containers. A logarithmic time axis is used to indicate the performance difference, which is in the microsecond range for the R-Tree and ND-Tree.

Collision Detection on a Single Page

The initial set of experiments involved a single-page device with varying component complexity. Fig. 4.17 shows that the R-Tree yields consistently better performance when no collisions occur because it does not examine elements on multiple levels; meanwhile, the ND-Tree suffers the overhead of graft construction which is not needed in a single-page setting. The Naï method lacks a comparable early exit strategy, and always executes the pairwise comparison to completion. The results for both the R-Tree and the ND-Tree exhibit volatile execution times due to increasing device complexity. As device density increases, the frequency of collisions increases.

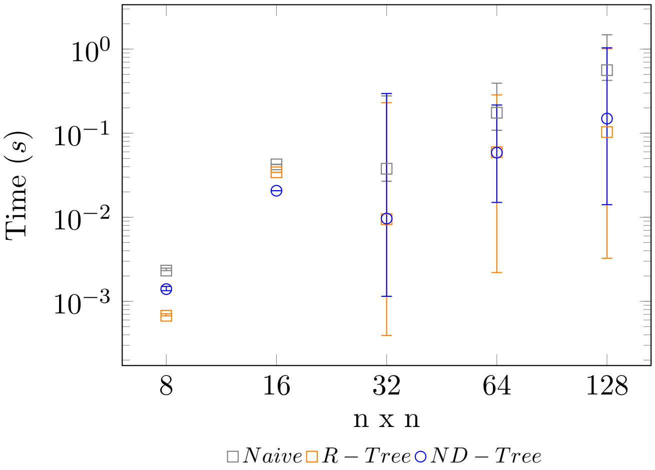


Figure 4.18: Runtimes where at least one collisions was detected between the devices located on a single page, yielding full tree traversal and direct collision detection between device elements. Full traversal and processing of the ND-Trees reached the processing of the actual device elements. A logarithmic time axis is used to accentuate the performance difference.

Fig. 4.18 reports the execution time when one or more collisions occur. Here, the ND-Tree performs nearly as well as the R-Tree as the ND-Trees processing gains required to handle the complexity of the devices overcome the construction overhead. For dense devices, the runtime of the ND-Tree can exceed that of the other methods due to the added calculations of the tree itself. Notably, the median performance of the ND-Tree continues to improve with increasing device density.

Collision Detection Over Multiple Pages

To test collisions across multiple pages, devices were constructed containing 128 components per page, with page counts ranging from 1 to 10, yielding a maximum of 1280 components. The runtime performance for multiple page devices are reported in Figs. 4.19 and 4.20. Fig. 4.19 reports the runtimes for the trials where no collisions occurred. As noted previously, the ND-Tree yields better performance as device complexity increases, but the addition of logical grouping at the graft level, allows for competitive performance with the R-Tree, despite its additional construction complexity giving the end user the ability to more tightly bind together fluid dynamic performance during layout and device construction.

In Fig. 4.20 reports the runtime for trials where one or more collisions occurred . Here, the ND-Tree not only yields better performance as device complexity increases, but the addition of logical grouping at the graft level allows for competitive performance compared to the R-Tree. The median runtimes are consistent with the R-Tree and the ND-Tree tends to outperform the R-Tree when comparing upper bounds. Lower bound performance still bears the brunt of the graft level overhead, but is negligible compared to

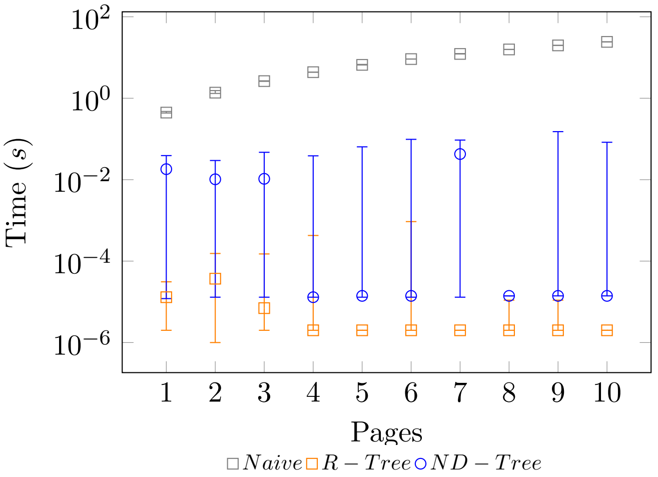


Figure 4.19: Runtimes for Naive, R-Tree, and ND-Tree for tests where no collision occurred among the devices. Devices consisted of 128 components on each page of each device from a single page to 10 pages total. The processing of the trees was limited to bounding containers only while the Naive method compared each component in one device to every other component in the other device. A logarithmic time axis was used to more clearly indicate the performance difference.

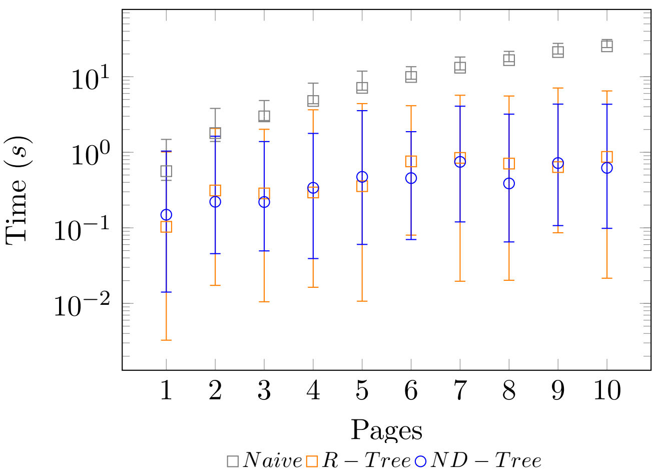


Figure 4.20: Runtimes for Naive, R-Tree, and ND-Tree for tests where collisions occurred among the devices. Devices consisted of 128 components on each page of each device from a single page to 10 pages total. The trees were traversed until all collisions were found while the Naive method simply checked for collision pairwise among all components. A logarithmic time axis was used to more clearly indicate the performance difference.

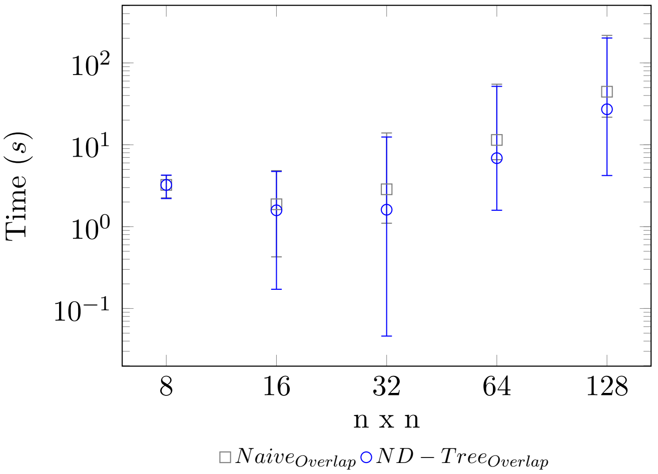


Figure 4.21: Runtime of collision detection between devices for both Naive and the ND-Tree methods, including the calculations needed to quantify the amount of overlap across all dimensions between colliding devices. The number of components listed were on each of 8 pages to gauge calculation response in the multilayer environment. The R-Tree does not facilitate quantification of overlap, and was therefore omitted from this comparison. A logarithmic time axis illustrates the performance difference.

the total runtime.

Additional Calculations due to Device Overlap

Fig. 4.21 reports the execution time of collision detection when performing the optional and computationally intensive calculations to quantify the amount of overlap that occurs. The R-Tree is incapable of quantifying overlap, therefore we omitted it from this experiment. Both the Naive and ND-Tree methods computed overlap in a consistent manner. The ND-Tree remains generally faster than the Naive method, because it eliminates the need to track non-colliding objects within devices. These observations correlate with collision detection results shown earlier in Fig. 4.18.

Chapter 5

Dynamic Radial Placement and Routing

In this chapter, a methodology is presented to dynamically place and route microfluidic components in a non-discrete design space where fluid volume usage, surface area utilization, and the timing required to perform specified biological assays are accounted for and optimized while also accelerating the development of potentially lifesaving new devices.

Many channel-based continuous fluid flow microfluidic systems have a linear layout in which fluid enters on one side of the chip and travels, under a pressurized flow, to the opposite side [95, 70]; in turn, many physical design algorithms targeting these technologies are based on a similar assumption [16, 100, 17]. Physical design algorithms for paper microfluidics should work with, rather than against, the natural radial flow of fluid in a porous medium; conversely, linear layouts, while simple to generate, are poor choices for paper microfluidic devices; which this argument experimentally confirmed.

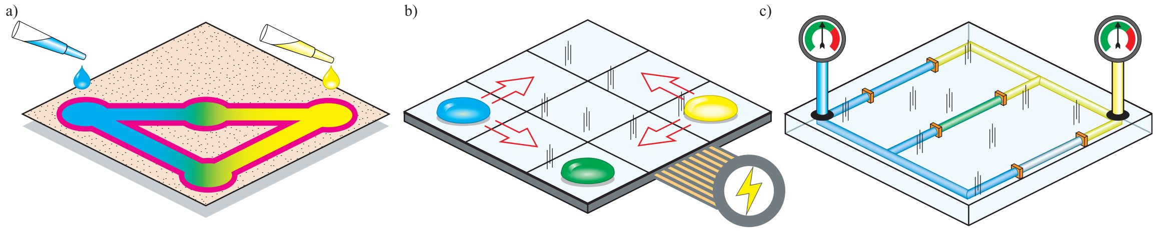


Figure 5.1: *A mixing operation using various platforms* a) On paper, fluid and reagents are delivered via pipette and fluid travels using capillary action bounded by printed wax barriers (in pink). Fluid channels are non-discrete so travel can occur in any planar direction. b) Electro-wetting utilizes an electric power source that controls electrostatic pads below a hydrophobic surface material to induce droplet travel. Consequently, droplets may only mode orthogonally from pad to pad. c) Pressurized channels and valves (represented by orange blocks) force fluid through embedded channels in a block of material that may either be machined with the channels, or 3-D printed.

Both pressure-driven flow through channels and capillary-force-driven flow through paper are convective flows. In the absence of any external force, such as gravity, fluid will flow equally in all directions (i.e., in a spherical direction in 3D or a radial direction in 2D). Channel barriers, which can be realized for paper microfluidic devices via wax printing, may constrain the otherwise uninhibited travel of fluid, but also introduce additional resistance to fluid flow [49]. Furthermore, evaporation, surface tension, and backpressure [64] place limits on the distance fluid can travel in a paper substrate.

Fig. 5.2 provides empirical evidence that linear layouts, i.e., those that might be generated by the algorithms described in Refs. [16, 100, 17] appropriately adapted for paper microfluidics, do not generate workable devices. Two linear paper microfluidic devices were fabricated, as depicted in the top portion of Fig. 5.2.

Both devices are simple branching routes where fluid is delivered to a large source reservoir so that it will flow toward a control sink at the opposite end. In both devices, twenty-four channels branch off from the main artery to divert the fluid to twenty-four sinks. In one of the two devices, the branches lie at a 90° angle to the main artery; in the other, the design was modified to reduce the angle of fluid diversion when entering the channels. In both of these tests, fluid only reached six of the twenty-four sinks, despite the fact that the fluid delivered to the source exceeded the calculated volume for the device and consequently encountered barrier failure. The second device, with less extreme fluid diversion, offered at best a marginal improvement in the volume of fluid delivered to the sinks, and likewise experienced overflow.

The second set of paper microfluidic devices, which serve to motivate this paper, are shown in the bottom portion of Fig. 5.2. In these "radial layouts," the source reservoir is placed at the center, and twelve sink reservoirs and one control reservoir were placed equidistant from the source. The radial layouts exploit tendency of fluid to flow in an expanding circle from the point of delivery, while the two linear devices shown on top aim to counteract the fluid's flow. In this experiment, the two radial layouts' twenty-four reservoirs were able to successfully fill, with less fluid, less paper area, and in less time than the two linear devices shown at the top of the figure.

The two devices shown on the bottom of Fig. 5.2 are smaller and have shorter channel lengths than the two shown on the top. As a matter of principle, similar device geometries could be laid out by appropriately adapting optimal or near-optimal physical design algorithms for continuous flow microfluidic chips that optimize these metrics [101, 36, 99, 113]. Fig. 5.3 illustrates one key difference between these algorithms and the approach presented here: existing physical design algorithms abstract away each component with a rectangular bounding box, and impose physical layout constraints that bounding boxes

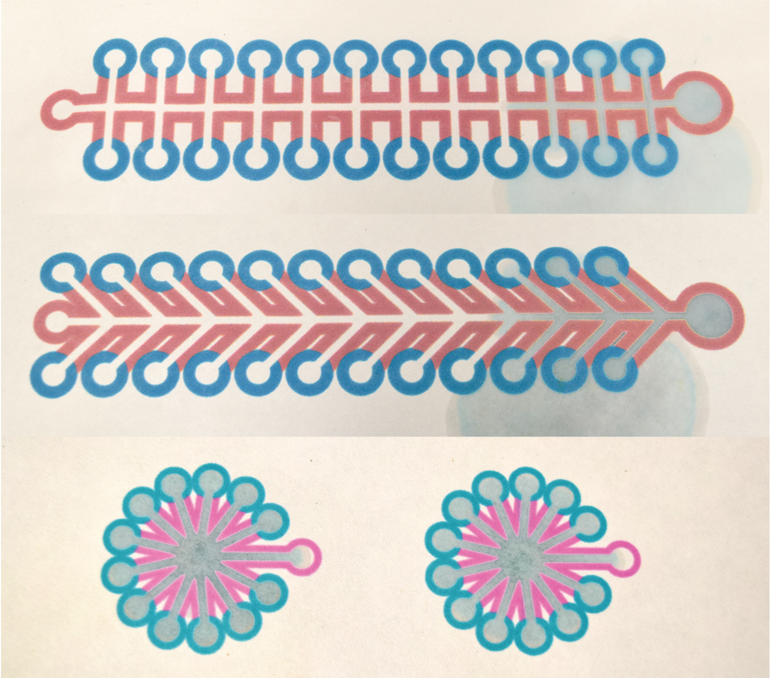


Figure 5.2: *Linear vs. Radial Layouts* (Top): A linear paper microfluidic device layout with 24 test reservoirs (blue), one control reservoir (red), a large source reservoir, all with 2 mm barrier widths. The device had a calculated volume of 117.4 μL therefore 140 μL fluid was delivered to the source reservoir. After 8 min 50 secs, only four reservoirs were filled, two additional reservoirs were partially filled, and the 2mm barriers ultimately failed. (Middle): A second linear device was fabricated but the channels were angled to aid in fluid flow. The change in channels increased the device's calculated volume to 137.5 μL, and therefore 160 μL fluid was delivered to the second device. After 9 min 32 secs, four reservoirs were again filled, two more reservoirs partially filled, and the device barriers failed. (Bottom): Two radial device layouts were made with 12 reservoirs (blue) plus 1 control reservoir (magenta), and only 1 mm barrier width. Using radial channels to aid in fluid flow, this device had only a 54.9 μL calculated volume, and consequently 54 μL fluid delivered to the left device and 50 μL to the right device. After 3 min 36 secs all 24 reservoirs filled completely and fluid reached both control reservoirs thereby successfully running to completion. The total fluid delivered to both bottom devices is less than the amount of fluid delivered to the top and middle devices, allowing two sets of tests to be performed using less fluid than the single test and without failure, demonstrating the advantage of a radial layout.

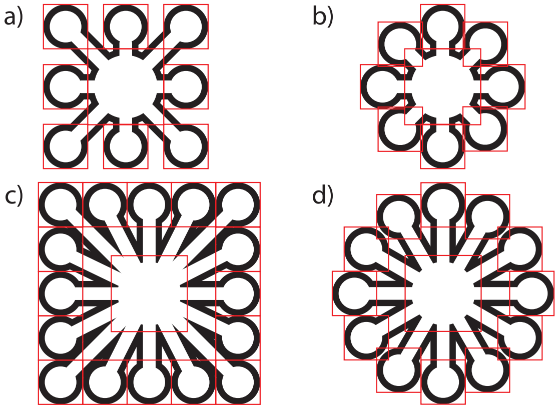


Figure 5.3: Using actual geometry versus abstract bounding boxes can significantly reduce fluid area and consequently materials and sample usage. Layout a) using bounding boxes for placement of components has a calculated volume of $284mm^2$ while b) has a calculated volume of $268mm^2$ yielding a 6% improvement over the bounds dominated version. Layout c) has a calculated volume of $618mm^2$, d) has a calculated volume of $482mm^2$ resulting in a 22% reduction of fluid area versus the bounds dominated layout.

cannot overlap, and that fluid channels cannot intersect bounding boxes unless they connect directly to an I/O port of the corresponding component. In contrast, the physical layout algorithm presented here detects component overlap based on component geometry, which is more accurate than the conservative bounding-box approach. As shown in Fig. 5.3 this yields tighter layouts and shorter routing channels.

The radial layout method presented here takes inspiration from the field of graph drawing. A radial tree (also called a radial map) draws a rooted tree by placing the root at the center of a circle and expanding the tree such that the levels are drawn on concentric circles [23, 7, 116]. It has been observed that the layouts produced by the algorithms for tree-shaped netlists do not resemble radial trees, but instead shares some principle similarities to H-trees [59], which were used in early multiprocessor interconnection networks [8] as well as VLSI clock tree routing [102, 9]. Radial tree drawing generalizes to radial graph drawing

[21], in which the vertices of a graph are drawn on a set of concentric circles; while this approach can place and route a netlist corresponding to any graph, and do not impose any constraints comparable to radial graph drawing. Additionally, this approach to channel route employs probes, taking inspiration from grid-based maze routers developed in the late 1960s [77, 41].

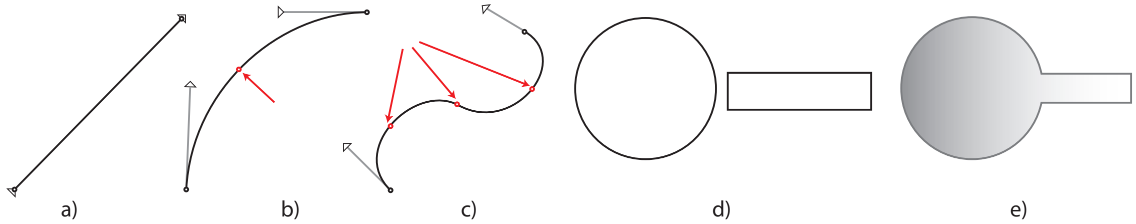


Figure 5.4: A Bézier curve is defined by 4 points in space: start and end points and 2 control points that define a parametric curve. The control points may or may not be on the curve itself. The curve may contain 2-5 critical points where it potentially changes direction, and possibly an inflection point where the curve changes direction. a) When the start and end points are also control points, the Bézier curve degenerates to a straight line. b) A curve defined by two control points, indicated by lines connecting them to the start and end point, and one maximum point (indicated by a red arrow. c) A curve with one maximum and one minimum point, along with inflection point between them where the curve changes direction (all three points indicated by red arrows. d) A series of Bézier curves defines a path; when the path starts and ends at the same point, a closed path creates a shape. e) Simple shapes can be joined to create more complex shapes.

Using the frameworks' library of paper microfluidic device components of reusable objects, rapidly assembled netlists form new diagnostic devices. Components are abstract elements that must reside in one and only one device, but may connect to multiple devices. A component definition may include functionality in terms of fluidic actions and abstract dynamics, such as mixing, transport, timing, etc. Each ***component*** $c = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ is physically defined by one or more geometric ***primitives***. A primitive ρ_i is constructed from one or more ***paths*** P_i, each of which using one or more ***Bézier curves*** (Fig. 5.4).

Bézier curves are parametric arcs defined with start and end points and "handle" points that constrain the curve. The curves may have 2-5 ***critical points***: 1) the start, 2) the end, 3) local maximum, 4) local minimum, and 5) an inflection point where the curve can change from concave to convex. The placement phase of the algorithm processes the critical points, as opposed to enumerating all sides, angles, and curves, in order to measure how close objects are placed to one another and to determine whether or not overlap occurs.

A ***device*** $D = \langle c_1, c_2, \ldots, c_n \rangle$ consists of at least one component, and encapsulates the desired actions and parameters needed to characterize its behavior. To create a device, individual components may be scaled or rotated as needed. Large devices may be specified hierarchically in terms of smaller devices, facilitating the concatenation of multiple assays in sequence or in parallel.

A ***netlist*** $N = \langle c_1, c_2, \ldots, c_n \rangle$ is a queue of components (Fig. 5.5) which determines the order in which components will be placed. Both components and devices may contain ***ports***, which define an interface for fluid transport. For example, a port on device D_1 may connect to a port on component c_1 $\{P_{D_1} \leftarrow P_{c_1}\}$; similarly, a port on device D_2 may connect to two ports on components c_2 and c_3 respectively $\{P_{D_2} \leftarrow P_{c_2,c_3}\}$. The physical location of a port within a component is defined as part of the component's specification. The physical location of a port within a larger device is not known until the physical location of the component containing that port has been placed within the device.

A ***layout*** $L = \langle D_1, D_2, ...D_n \rangle$ (Fig. 5.5) contains all devices residing on one or more ***pages*** and also defines environmental variables such as substrate type, composition, and size, temperature, humidity, and other variables as defined by the end user. Once a

Table 5.1: Variables used in the Algorithms

Vars	Description
c	Component to be placed
cf	The cost factor of Component c placed in relation to one or more previously-placed Devices D
Chs	A set of Channel Components
ch	A Channel Component
D	A Device that contains one or more Components that have been been placed and possibly routed
Δ_δ	The set of minima and maxima for each dimension
I	A set of intersections
L	A Layout containing one or more fully laid-out Devices
N	The netlist; a set of Components to be placed ordered as collections of sources to sinks
O	The path that serves as an outline to a Device or Component
p	A point (x, y)
P	A Path object consisting of one or more Segments s
P_D	A sub-path of the outline of D which consists of all its critical points and the angle about the center each are located
P_c	Similar to P_σ but for a component c
$Page$	Contains the finished Layout L along with information about substrate properties
Φ	Port variable that describes the input (x_i, y_i) and output locations (x_o, y_o) between Components and/or Devices
Q_ρ	A priority queue consisting of the x, y coordinates and rotation θ of Component c to be placed that is sorted on cost factor cf
s	A Segment object defined by a Bézier curve
Θ	The aperture $\langle\theta_\alpha, \theta_\omega\rangle$ of exposure between a component c that is the starting angle θ_α and the ending angle θ_ω about its center point
$\theta_{\alpha,\omega}$	The start and stop angles of an object that defines an arc angle
α	the starting value
ω	the ending value

layout is completed, the framework renders the device using established file formats, such as PDF, DXF, and SVG.

5.1 Physical Design Algorithm

The approach to paper microfluidic physical design is to work radially outward from source fluid reservoirs to sink reservoirs while seeking to maintain the minimum distance that fluid must travel. Components are placed one-at-a-time. Potential locations for each new component are enumerated by a 360° sweep, motivated by the way that a radar screen displays information. At each potential location, the component may also be rotated 360° to best fit the component into the subset of the device layout that has been generated thus far.

In this manner, a listing of potential placement locations is sorted by how closely they abut the existing layout, while minimizing any desired parameters such as shortest critical path, fluid volume, time to complete, etc., as secondary criteria. A route is computed for each potential location, based on the premise that the closest positions are likely to have the shortest routes, although no such claim can be guaranteed in the general case. If a suitable route is found, the component is permanently placed and connected to the layout. The algorithm terminates preemptively if all other candidate placement locations are not routable; it terminates successfully when successful placement locations are found for all components.

Algorithm 2 The Radar Place and Route Algorithm

```
function Radar($N$)
    $D \leftarrow$ new Device()
    $L \leftarrow$ new Layout($D$)
    while $c \leftarrow N$.pop()
        $Q_\rho \leftarrow$RadarPlace($D \in L, c$)
        if $Q_\rho.empty()$
            return $L$
        else
            while !$Q_\rho.empty()$
                $c.(x, y, r) \leftarrow Q_\rho.pop()$
                if !RadarRoute($D, c$)
                    return $L$
                end if
            end while
        end if
        Merge($L, c, Chs$)
        $L.center \leftarrow Page.center$
        if $L.w > Page.width$ —— $L.h > Page.height$
            return $L$
        end if
    end while
    return $L$
end function
```

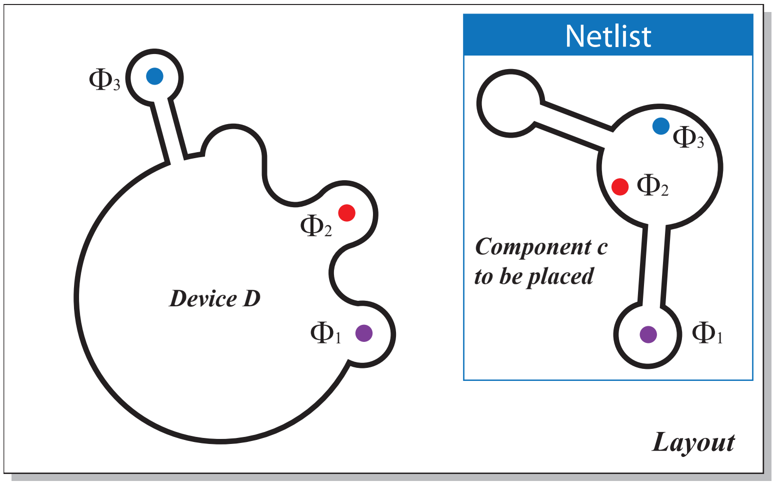


Figure 5.5: The Radar algorithm generates a ***layout*** of one or more ***devices*** constructed from a ***netlist*** of ***components***. *Components are selected one-by-one for placement; after each component is placed, it is connected to the existing layout by* ***channels*** *that route fluid throughout the layout.*

5.1.1 The Radar Algorithm

The Radar algorithm (Alg. 2, Fig. 5.5) takes as input a netlist of components and, optionally, a buffer value, which is the minimum allowable distance between components after placement. At the onset of the algorithm, the layout L is initialized with the environmental parameters determined by the user and an empty device to be constructed from the netlist. Table 5.1 lists and briefly describes all variables and data structures used. The algorithm does not concern itself with the viability or the nature of the component during placement, but does check for whether or not the component is a source, sink, or internal, to determine whether or not routing needs to be performed. A source is a component with no input channels, and a sink is a component with no output channels; the netlist can have any number of source and sinks. At the start of each pass, the next component $c \in N$ is popped from the netlist and attempted to be placed and routed. A priority queue

Q_ρ (Table 5.1) is initialized to hold the set of potential locations and orientations for c that RadarPlace will attempt to generate when called. Q_ρ is ordered by increasing ***cost factor***, a value calculated for each candidate position that is calculated using a user-defined method. The default approach is to only consider total surface area or fluid volume but more complex methods can be considered, depending on the user's choice of optimization criteria. Should the list be empty, no suitable placement was found and the algorithm terminates. Otherwise, c is set to the calculated values of each candidate placement in Q_ρ until either a valid routing for c is discovered, or Q_ρ is emptied, resulting in a failed routing. The final placement and route of each component is then merged into L using the Merge function for the next pass. The device layout is then centered in the page layout and evaluated to determine if extending beyond the page dimensions and returned if exceeded with an error. If failure occurs at any point (Fig. 5.11), or N is emptied and the algorithm completed, the finished or currently constructed layout L is returned and flagged with any appropriate errors.

5.1.2 The RadarPlace Algorithm

RadarPlace (Alg. 3) generates an ordered positional and orientation queue Q_ρ for evaluation of potential placements for component c in layout L sorted on CostFunction results. The sets P_D and P_c contain tuples of critical points of the Bézier paths for $D \in L$ and c respectively. The ***aperture*** $\Theta\langle\theta_\alpha, \theta_\omega\rangle$ is defined to be the start and end angles to be checked for either D or c and is initialized to a full 360° sweep. P_D obtains the results of scanning D performing a full sweep of its border using an algorithm called CurveScan

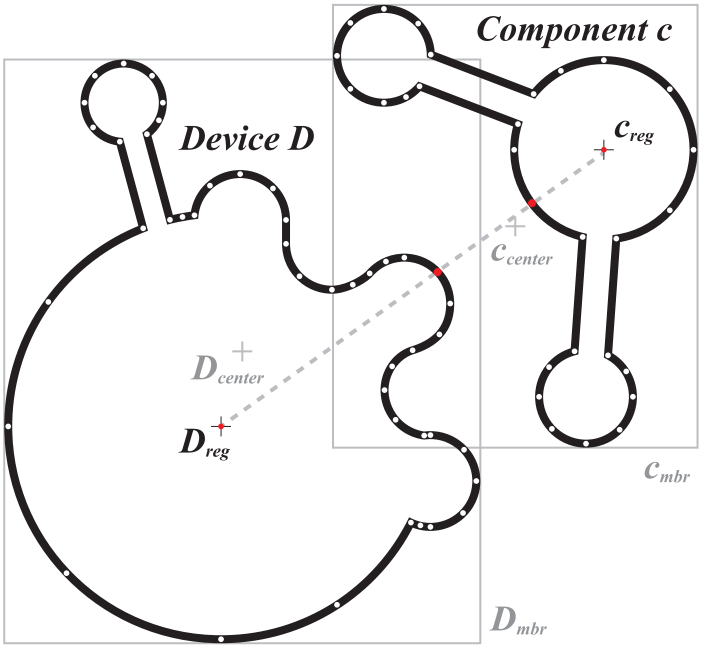


Figure 5.6: A component c whose placement and orientation will be determined relative to a device D comprising one or more previously-placed components. The objective is to minimize the sum of the distances between each pair of critical points in c and D, which also aligning with either the center points or the registration points of each object as chosen by the user. The two center points c_{center} and D_{center} are defined as the mid-point vertically and horizontally of the minimum bounding rectangles, c_{mbr} and D_{mbr}, respectively. The registration points c_{reg} and D_{reg} are defined as local origin points $(0, 0)$ for which all measurements within D are calculated and also where the location of D in L is determined. In this example, c and D have 29 and 44 critical points, respectively, resulting in 1276 distance measures for ***each*** candidate location.

(Alg. 4 and Section 5.1.2). CurveScan returns the set of critical points that c will be compared against during the **for** loop spanning lines 7-25 of Alg. 3 and depicted in Fig. 5.6.

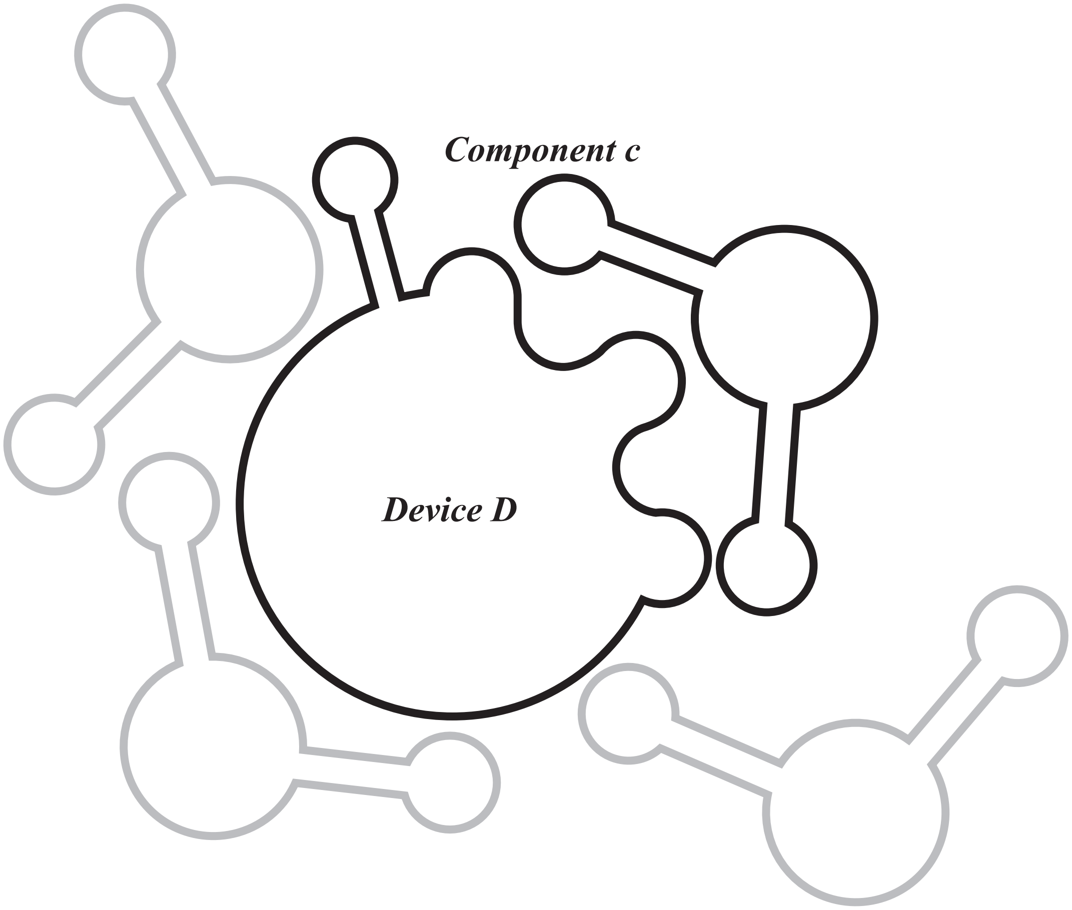


Figure 5.7: When there are no ports to be connected, RadarPlace rotates the component c up to 360° for each candidate location to compare all critical points of c to all critical points of placed devices $D \in L$.

Algorithm 3 The RADAR Place Algorithm

```
function RADARPLACE($D, c$)
    $Q_\rho \leftarrow \emptyset$
    $\Theta\langle\theta_\alpha, \theta_\omega\rangle \leftarrow (0^\circ, 360^\circ)$
    $P_D \leftarrow$ CURVESCAN($D_{outline}, \Theta$)
    for $p_i \in P_D$
        $c.(x, y) \leftarrow p_i.(x, y)$
        $c.moveby$(COLLIDE($D, c$))
        if $c.connected$
            $\Phi_D \leftarrow D.ports.at(\lceil D.ports.size/2\rceil)$
            $\Phi_c \leftarrow c.ports.at(\lceil c.ports.size/2\rceil)$
            $c.rotate$(GETPORTANGLE($\Phi_D, \Phi c$))
            $c.moveby$(COLLIDE($D, c$))
            $\Theta\langle\theta_\alpha, \theta_\omega\rangle \leftarrow$ VIEWWINDOW($D, c$)
        else
            $\Theta\langle\theta_\alpha, \theta_\omega\rangle \leftarrow$ VIEWWINDOW($D, c$)
        end if
        $P_c \leftarrow$ CURVESCAN($c_{outline}, \Theta$)
        for $p_j \in P_c$
            $\theta_\tau \leftarrow$ GETANGLE($D.center, c.center$)
            $\theta_\sigma \leftarrow$ GETANGLE($p_i.(x, y), c.center$)
            $c.rotate((\theta_\sigma - \theta_\tau), c.center)$
            $c.moveby$(COLLIDE($D, c$))
            if COLLIDE($D, c$) $== 0$
                $Q_\rho.add(\langle c.(x, y), c.rotation,$ COST(D,c)$\rangle$ )
            end if
        end for
    end for
    return $Q_\rho$
end function
```

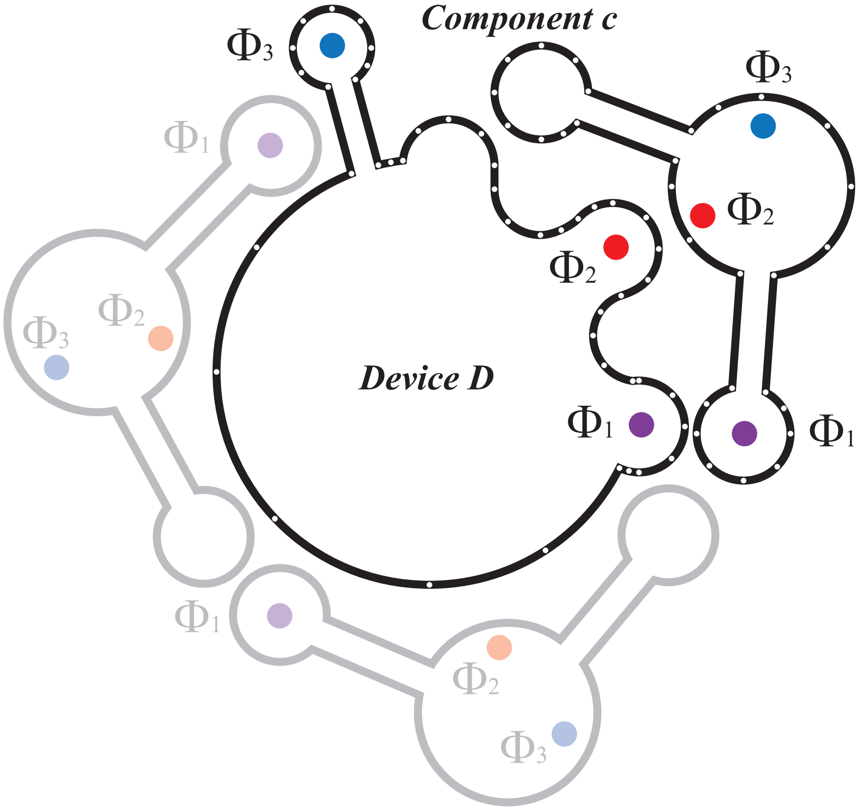


Figure 5.8: When there are ports to be routed, RadarPlace restricts the rotation of c to ensure that the ports are oriented toward the points in D to which they connect.

CurveScan

The CurveScan algorithm (Alg. 4) generates a sub-curve comprising a set of points contained in a Bézier path O bounded radially by a start/ending angle Θ, and returns the sub-path outline of the object bounded by those angles for analysis, as illustrated in Fig. 5.9. Each segment s within the sub-path is examined to determine if it falls within the aperture Θ, which is defined as the angle formed between the edges that are reachable from the source object to destination object, similar to the visible face of the moon when viewed from earth. If so, the curve and any connecting curves are added to sub-path P if not already present in P. CurveScan increments the angle by the $radianInterval$ formed by the last curve added minus the current angle and terminates after examining all segments in O and returns P within the aperture Θ.

Placing Components

At the start of each iteration (Line 7 of Alg. 3), c is initialized so that it is placed such that the center point of c is set to the same coordinates as critical point p_i to initialize the location of c to have some overlap with the already placed items in L. The Collide function computes the amount of overlap between c and D. If the amount of overlap is non-zero, then this information is used to move c to eliminate overlap currently in L. For example, should Collide return $\{15.0, -5.0\}$, c would be moved 15 units to the right along the x-axis, and 5 units down the y-axis.

Depending on whether c is connected to the current layout or not – meaning a new source – the algorithm then calculates the aperture of points to be considered from D and

Algorithm 4 The CurveScan Algorithm

```
function CurveScan($O, \Theta_{\alpha,\omega}$)
    $P \leftarrow \emptyset$
    if  $\Theta_\alpha$ undefined
        $\Theta_\alpha \leftarrow 0$
    end if
    if $\Theta_\omega$ undefined
        $\Theta_\omega \leftarrow 360$
    end if
    while $\Theta_\alpha < \Theta_\omega$
        $line \leftarrow (O.center(x,y), \Theta_\alpha)$
        $I \leftarrow$ GetIntersections($O, line$)
        for $i \in I$
            $s_{temp} \leftarrow s_i$ intersected farthest from $O.center$
            if ( $!s_{temp} \in P$ )
                $P$.add($s_{temp}$)
                if ( !( ($s \in O_{i-1}$) $\in P$ ) )
                    $P$.add($s \in O_{i-1}$ to $s_{temp}$ )
                end if
            end if
        end for
        $\Theta_\alpha \leftarrow \Theta_\alpha + radianInterval$
    end while
    return $P$
end function
```

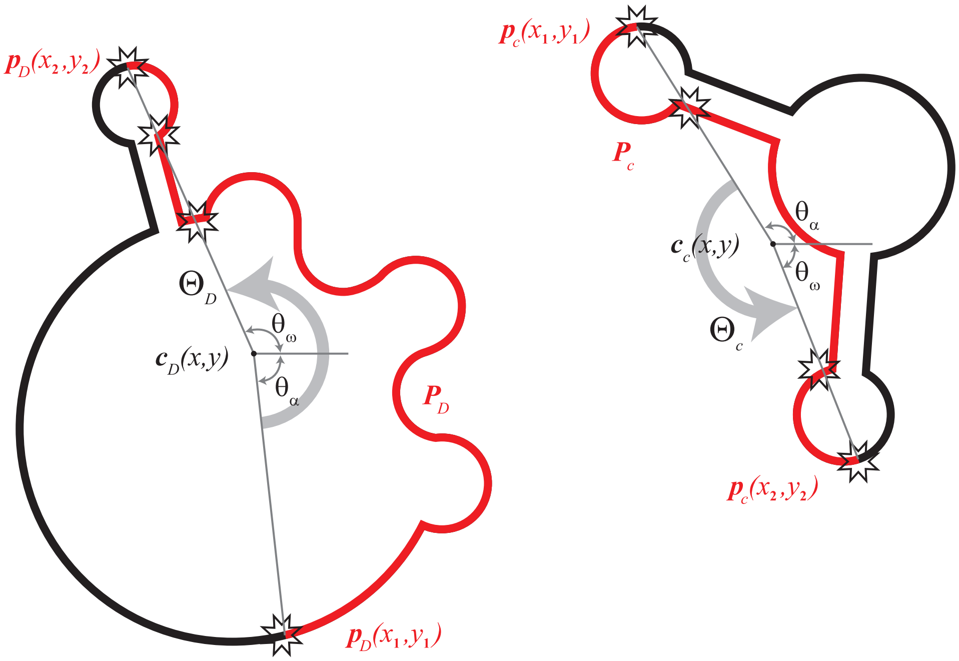


Figure 5.9: CurveScan retrieves a sub-path P_D from a device's outline or P_c from a component's outline based on a start and end angle $(\theta_\alpha, \theta_\omega)$, which were determined by ViewWindow (Alg. 4 and Fig. 5.10). The start and end points are calculated in standard polar orientation in a counter-clockwise manner with the rotational origin at the center point of the outline and 0° aligning with the standard x-axis orientation. The right component's sub-curve is determined by the furthest point intersecting with the start angle $P_c(x_1, y_1)$ (e.g. 110°) and end angle $P_c(x_2, y_2)$ (e.g. 270°). The left sub-curve begins earlier in the rotation (e.g. 290°) but is internally converted to a negative value (e.g. −70°) to return the proper sub-curve from points $P_D(x_1, y_1)$ to $P_D(x_2, y_2)$.

c when calculating the distance between each candidate location to place c to the closest non-overlapping object in L. If c has no ports to connect (Fig. 5.7), all points of c are compared against all points of all devices in L. The base case assumes c is connected and therefore retrieves the middle connecting port pair(s) from D and c based on their indexed location in an internal *ports* array in c. Port specification is determined by the user prior to algorithm execution and therefore the order of indexing is fixed. The component c is rotated to align the source in D to the sink in c for the purpose of minimizing channel crossing in the routing phase, as shown in Fig. 5.8 By aligning the middle ports, the most direct ports would be connected initially and then connected to each side in turn. In each case, ViewWindow is called to obtain the rotational angle of arc aperture (Table 5.1) for both D and c and is assigned to Θ for passing into CurveScan (Alg. 4). CurveScan then returns the points P_c on c that are to be considered in the inner **for** loop.

Determining Aperture with ViewWindow

It is necessary to calculate the viewing aperture from the bounds of one component to the bounds of another component determined from their respective centers as shown in Fig. 5.10. ViewWindow takes in two objects – devices or components – and begins by identifying the angle θ_τ between their centers $c_\sigma.(x.y)$ and $c_\tau.(x.y)$. θ_λ and θ_ρ are the perpendicular angles to that angle which are used to find the outermost intersections points to find the start and end points $p_\sigma.(x,y)$ and $p_\tau.(x,y)$ of the profile on each component. The angle formed between those points for each component $\Theta.\alpha$ and $\Theta.\omega$ is then returned in the aperture Θ.

Algorithm 5 The ViewWindow Algorithm

```
function ViewWindow($c_\sigma, c_\tau$)
    $\Theta.(\alpha, \omega) \leftarrow (0, 0)$
    $\theta_\tau \leftarrow$ GetAngle($c_\sigma.(x, y)$, $c_\tau.(x, y)$)
    $\theta_\lambda \leftarrow \theta_\tau + 90°$
    $\theta_\rho \leftarrow \theta_\tau - 90°$
    $p_\sigma.(x_1, y_1) \leftarrow$ Intersect($c_\sigma$, $\theta_\lambda$)
    $p_\tau.(x_1, y_1) \leftarrow$ Intersect($cC_\tau$, $\theta_\lambda$)
    $\Theta.\alpha \leftarrow$ GetAngle($p_\sigma.(x_1, y_1)$,$p_\tau.(x_1, y_1)$)
    $p_\sigma.(x_2, y_2) \leftarrow$ Intersect($c_\sigma$, $\theta_\rho$)
    $p_\tau.(x_2, y_2) \leftarrow$ Intersect($c_\tau$, $\theta_\rho$)
    $\Theta.\omega \leftarrow$ GetAngle($p_\sigma.(x_2, y_2)$,$p_\tau.(x_2, y_2)$)
    return $\Theta$
end function
```

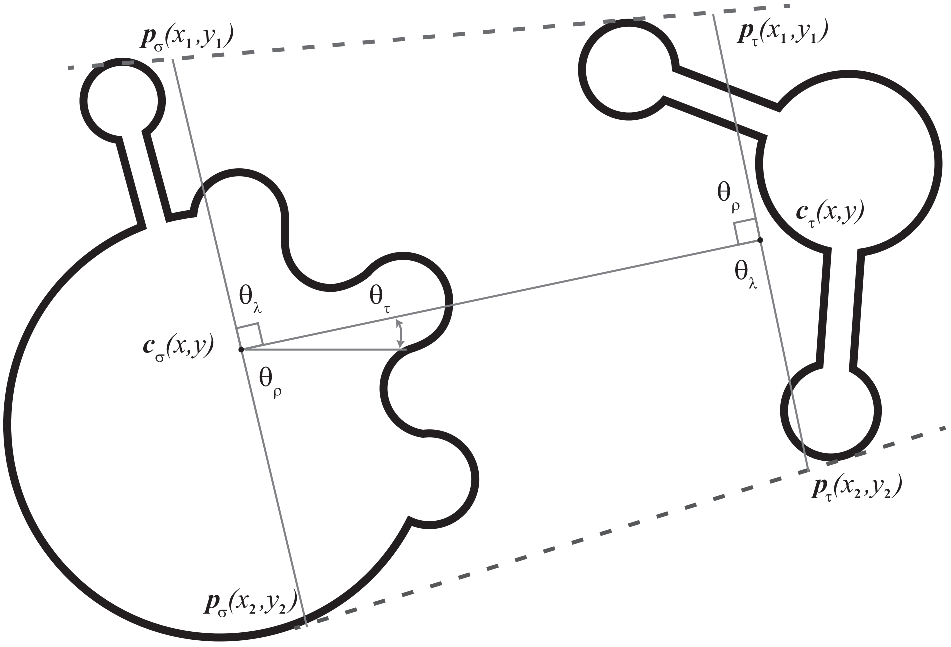


Figure 5.10: ViewWindow identifies the profiles of the devices D in layout L that would face component c when drawing lines from the outer extrema of each device. This determines the portion of the objects' profiles that need to be considered when calculating the closeness of the objects to one another. Devices and components do not have to be connected to anything currently in the layout.

Comparing Critical Points

For each point pair, c is rotated to align p_j and p_i and their centers of rotation (Fig. 5.6). The rotation angle is the difference between the angles formed from the line connecting the centers of the place devices in L and c, as well as the line connecting p_i and the center of c. c is then moved so that its center point is equal to the current critical point p_i, which ensures that overlap will occur. Next, c is shifted by the amount of overlap to a location expected to be outside the bounds of all devices in L. Should the new location also result in an overlap as shown in Fig. 5.11, then the location is discarded and the loop continues. If there is no overlap, the location, orientation, and cost factor of the location is added to the priority queue. Once all desired points have been evaluated, Q_ρ is returned.

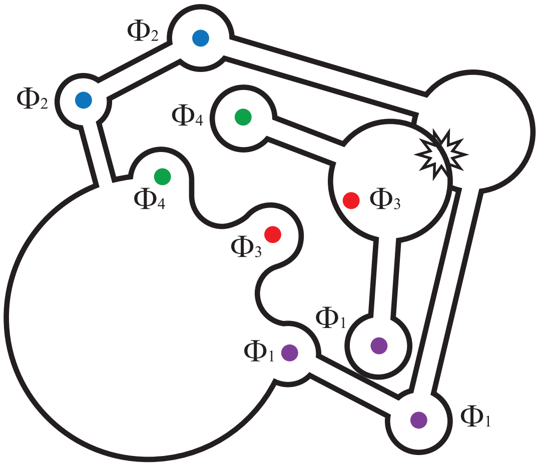


Figure 5.11: In this example, RadarPlace is unable to place c without colliding with already placed components in L and the candidate location is discarded from consideration.

Evaluating the Cost of Potential Locations

The placement phase of the algorithm requires a method of determining what is "best" when identifying potential placements for components. The Cost function in Alg. 3 is an abstracted method call that returns a numeric value which determines its rank in the priority queue which sorts the candidate placements by that rank. Paper microfluidic devices have several mitigating factors that affect performance of the fluid in the substrate and depending on the specific application the end user is targeting, the relative importance of each of these factor may alter what constitutes an optimal and/or effective device.

Several mathematical models exist [63] that characterize various physical properties that an end user may want to employ in determining a cost factor for a component location. The Lucas-Washburn model (eq. 5.1) applies to one-dimensional flow and calculates length of travel of a fluid over a particular time – which is useful when seeking a method to minimize channel lengths

$$l = k\sqrt{\frac{\sigma}{\mu}t} \tag{5.1}$$

where "k is a proportional constant, σ the surface tension and μ the viscosity of liquid, and t time. The proportional constant k depends on the material properties of the porous medium including the pore diameter, contact angle between the liquid and porous medium, and tortuosity of the porous medium." [37] Alternatively, another useful model [42] that accounts for capillary action and the fluid interaction with channel boundaries might be

$$l(t) = \alpha\sqrt{\left(1 + \beta \frac{d}{\phi^{\frac{1}{3}} w} \frac{\cos\theta_b}{\cos\theta}\right) \frac{\sigma}{\mu} t} \tag{5.2}$$

where α is an empirical co-efficient based on experimental results, β is a correction co-efficient, W is the channel width, θ_b is the contact angle with the barriers, θ is the contact angle with the substrate, ϕ represents the porosity value of the substrate, σ the vapor-liquid interfacial tension, and d the diameter of the capillary tubes in the medium. The dynamics of modelling and optimizing cost factors in a paper microfluidic setting can greatly increase the algorithmic complexity and computational overhead. While, the cost functions can be determined by the paper microfluidic device designer based on the desired metrics for optimization, the default approach, which is presented here, utilizes the paper area, and, by extension fluid volume, as the primary objective. Specifically, this approach tries to minimize the distance between placed components, and calculates how close they are to one another, as discussed in the following subsection.

Quantifying "Closeness" with the Snugness Factor

The concept of snugness when evaluating relative placement of components is defined as the minimal amount of space between 2 objects. The HowSnug algorithm (Alg. 6 takes in two Bézier paths and calculates the area between them which represents the concept of snugness between components. Using the components passed in, ViewWindow is called to determine the aperture of exposure between the components which is then passed into CurveScan which returns the sub-path from the outline of each component which forms the paths where the area between is calculated.

Algorithm 6 The HowSnug Algorithm

```
function HowSnug($D, c$)
    $\Theta \leftarrow$ ViewWindow($D, c$)
    $P_D \leftarrow$ CurveScan($D, \Theta$)
    $P_c \leftarrow$ CurveScan($c, \Theta$)
    $(x_2, y_2) =$ Max($P_D.p_1, P_D.p_n, P_c.p_1, P_c.p_n$)
    $(x_1, y_1) =$ Min($P_D.p_1, P_D.p_n, P_c.p_1, P_c.p_n$)
    $Box.area = (x_2 - x_1) * (y_2 - y_1)$
    $P_D.area \leftarrow$ BézierArea($P_D, box.(x_1, y_1)$)
    $P_c.area \leftarrow$ BézierArea($P_c, box.(x_2, y_2)$)
    $SnugFactor \leftarrow (Box.area - P_D.area - P_c.area)$
    return $SnugFactor$
end function
```

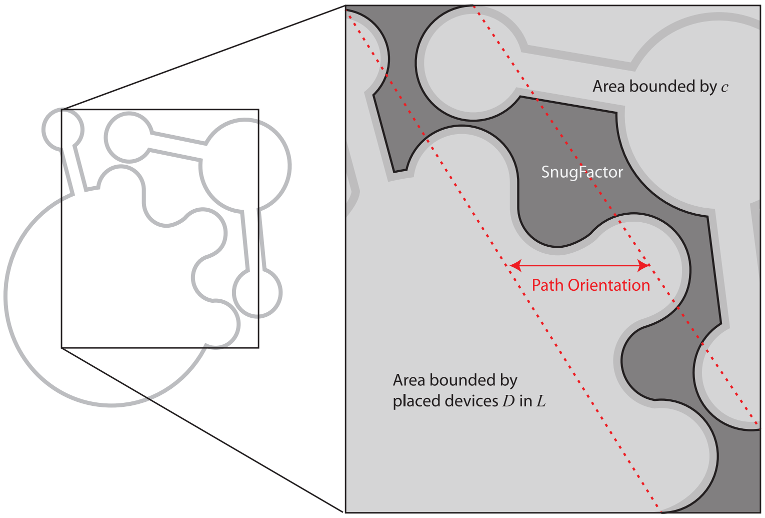


Figure 5.12: The snugness factor of two curves, as computed by HowSnug. The snugness factor is defined to be the area between the two curves within a bounding box. The path orientation (in red) cannot be vertical; if this occurs, both paths are rotated 90° prior to computing the snugness factor.

Figure 5.12 illustrates computation of the snugness factor. The first step is to generate the smallest bounding box that contains both curves. The start and end points of both sub-curves P_D and P_c are compared to find the maximum and minimum points of the bounding box and the $Box.area$ is calculated using these points. Second, the orientation of the curves is determined; if the orientation is vertical, the bounding box and curves are rotated 90°. Third, the area under both curves, but within the bounding box, is calculated using a standard polygon decomposition method [53]; let $P_D.area$ and $P_c.area$ denote these areas. Then the snugness factor is computed as $Box.area - P_D.area - P_c.area$.

MinMaxDelta

Designers of real-world paper microfluidic devices often need to *minimize* some values while *maximizing* others. For example, consider a home pregnancy test, probably the most common example of paper microfluidics. These tests consist of a strip of paper; urine is applied to one end, and capillary action transports the urine past two or more test lines to an absorbent pad on the other end of the strip. The test lines change color when exposed to human chorionic gonadotropin (HCG), a protein present in the urine of pregnant women. It is advantageous to *maximize* the amount of urine that passes through the test lines, because more urine means more HCG detected (and therefore a more-pronounced color change and an easier-to-read test result). Designers accomplish this by maximizing the size of the absorbent pad, which functions as a pump to drive urine flow past the test lines. It is also advantageous to *maximize* (up to a practical limit) the run time of the assay (the amount of time spent flowing urine past the lines). Simultaneously, it is advantageous to *minimize* the size of the paper strip, since it only serves to conduct urine through the test lines, and minimizing overall device size reduces costs associated with device fabrication, packaging, and shipping.

The function MinMaxDelta supports maximization and minimization optimizations like these. The function computes a running tally of the minimum and maximum differences in value for each dimension among a set of dimensions passed in as a parameter. It returns the appropriate minimum or maximum value in either the positive or negative directions along each dimension. For example, assuming a running tally of surface area (that is to be minimized) and runtime (that is to be maximized) $S_\Delta = \{10.0_{min}, 15_{max}\}$. The algorithm has determined there is a potential placement with a value of $T_\Delta = \{9.0_{min}, 10_{max}\}$ and when fed into MinMaxDelta the resulting values are $S_\Delta = \{9.0_{min}, 15_{max}\}$. Alternatively if $T_\Delta = \{21_{min}, 10_{max}\}$, then S_Δ would not change; or if $T_\Delta = \{5_{min}, 20_{max}\}$ then it would have $S_\Delta = \{5_{min}, 20_{max}\}$.

5.1.3 Radar Route

RadarRoute (Alg. 7) is invoked when a candidate location has been identified for a component c and at least one of c's ports has been matched with corresponding port on the device D in the current layout L. RadarRoute attempts to route one or more channels to deliver fluid from D to c. The number of port pairs between c and D determines the number of routes required.

The algorithm initializes sets to hold Channels Chs constructed during the algorithm and the intersections of the probes I (Line 2). The $buffer$ value is set to 1/2 the resultant width of the channel to be constructed (Line 3). In Line 4, the number of port pairs is determined to be odd or even and then the middle index value of the port pairs is chosen (Line 5) and is used as the initial iterative values for i and j (Line 6). Finally, if the port pair count is even, j is incremented to the next port pair index value (Line 7). The example in Fig. 5.13 shows three port pairs $\{\Phi_1, \Phi_2, \Phi_3\}$ to be connected by three channels. The port pairs are ordered for routing from the middle toward the two perimeter of the device. As shown in Fig. 5.14, the routing order is $\{\Phi_2, \Phi_1, \Phi_3\}$.

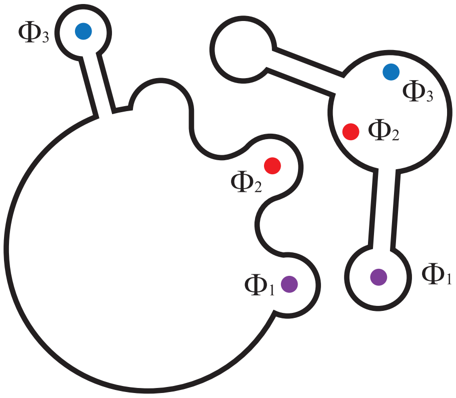


Figure 5.13: RadarRoute starts with component c already placed and 3 port pairs to be connected: Φ_1, Φ_2, and Φ_3. Connections are routing starting with the middle port pair Φ_2, followed by Φ_1 and Φ_3 to reduce the likelihood that routes cross.

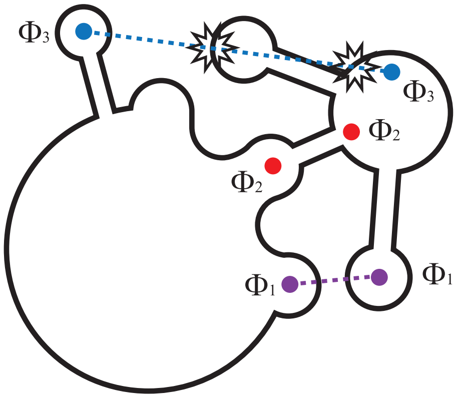


Figure 5.14: The probe generated for port pair Φ_2 is unobstructed, so the corresponding channel is routed. Port pair Φ_3's probe collides with the outline at two distinct collision points.

Algorithm 7 The Radar Route Algorithm

```
function RadarRoute($D, c$)
    $Chs \leftarrow \emptyset$, $I \leftarrow \emptyset$
    $buffer \leftarrow channel.width/2$
    $odd \leftarrow$ isOdd($c.inputs.size$)
    $midpoint \leftarrow \lceil c.inputs.size/2 \rceil$
    $i, j \leftarrow midpoint$
    $(odd)$ ? $j \leftarrow i$ : $j \leftarrow i + 1$
    while $i >= 0$ —— $j < c.inputs.size$
        $p_{D_i}.(x, y) \leftarrow D.\Phi_i.(x, y)$
        $p_{D_j}.(x, y) \leftarrow D.\Phi_j.(x, y)$
        $p_{c_i}.(x, y) \leftarrow c.\Phi_i.(x, y)$
        $p_{c_j}.(x, y) \leftarrow c.\Phi_j.(x, y)$
        $probe_i \leftarrow$ new Path($p_{D_i}(x, y), p_{c_i}(x, y)$)
        $probe_j \leftarrow$ new Path($p_{D_j}(x, y), p_{c_j}(x, y)$)
        for each $probe_{i,j}$
            if $I \leftarrow$ Intersect($c, probe$)
                $probe.addNode(I.node_\alpha)$
                $probe.addNode(I.node_\omega)$
                $curve \leftarrow$ GetSubCurve($c, I$)
                $precurve \leftarrow$ GetSubCurve($probe, I_\alpha$)
                for each $node \in curve$
                    $probe.addNode(node)$
                    Move($probe.node, buffer$)
                    if Intersect($curve, precurve$)
                        return $Chs$
                    end if
                end for
            end if
            if HowSnug($D, c$) $< channel.width$
                $c.moveby(channel.width)$
            end if
        end for
        if $odd$ && $i == midpoint$
            if $probe_i.length < probe_j.length$
                $Chs.add($ new Channel($probe_i$) )
            else
                $Chs.add($ new Channel($probe_j$) )
            end if
            $j \leftarrow i$
        else
            $Chs.add($ new Channel($probe_i$) )
            $Chs.add($ new Channel($probe_j$) )
        end if
        Merge($c.outline, Chs$)
        $i \leftarrow i - 1$
        $j \leftarrow j + 1$
    end while
    return $Chs$
end function
```

While there are channels to be routed, the algorithm pulls the port coordinates pairs (Fig. 5.13: $\{\Phi_1, \Phi_2, \Phi_3\}$) that will serve as the abstraction that will attempt to connect the two locations on D and c (Lines 9-12). A Bézier path for each connection is instantiated to serve as the "probe" connecting the source and sink locations (Lines 13-14). If the probe intersects with c (Fig. 5.14) (Line 16), the intersection points are added to a set along with the points that make up the sub-curve of c between them (Lines 17-19). The *precurve* (Line 20) is sub-curve of the start of the *probe* to the point where it intersects c and is used to determine if the probe intersects with itself.

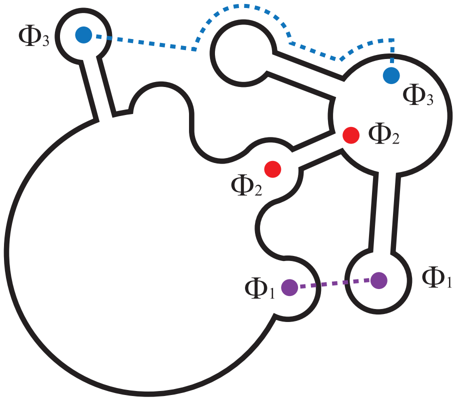


Figure 5.15: A sub-path is routed around the collision ports arising from port pair Φ_3's probe. The sub-path follows the contour of the new component that is added to the layout.

The points in the set *curve* are then each evaluated by creating a new node in the *probe* (Line 22) corresponding to the original node in c and moved away from c by half of the width of the channel component that will be generated (Fig. 5.15, Line 23). In this way, the *probe* will produce a curve running parallel to the previously intersected sub-curve of c. Based on the current position of c with respect to L, c may be moved to a distance away from L to allow for the width of the channel to be routed (Fig. 5.16) if c

collides with any already-placed components. The sub-curve is also tested for intersection with the *precurve* (Lines 24-25); a positive answer here would indicate that the probe under construction intersects with itself (Fig. 5.17); if this occurs, the probe cannot complete the route and the algorithm terminates. On Line 26, a quick check of the placement of c's snug factor will determine if c needs to be moved away from D by the width of the channel before channel construction begins.

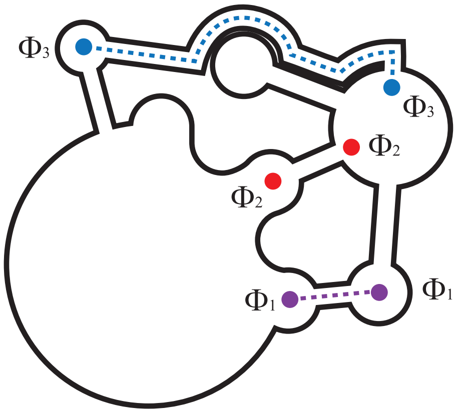


Figure 5.16: The routed probe is adjusted to an offset equal to half of the width of the channel being routed, plus any desired buffer distance between barriers of distinct channels. This routes the channel to connect port pair Φ_3. Port pair Φ_1 is also routed trivially.

The algorithm then handles the initial case of routing: where the starting channel is either the middle value of an odd number of channels or if there is an even number. The concern only occurs on the first route as all subsequent routings are performed as left /right pairs. Therefore in the initial case where the number of channels is odd, 2 probes are run with the same source and sink but are deflected each to the left and right (Lines 28-36). At completion of probe generation, the probe length is evaluated and the shorter probe is used to generate the Channel.

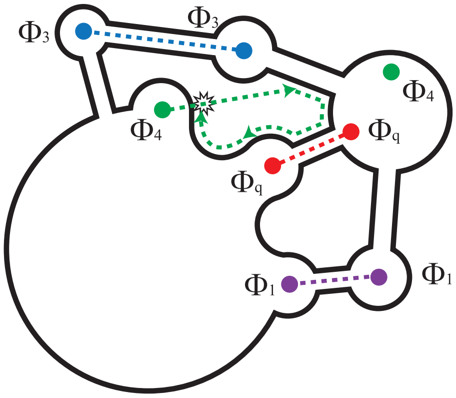


Figure 5.17: Example of a probe failure. In this case, in attempting to connect port pair Φ_4, the probe loops back and collides with itself indicating the probe is trapped by already placed components.

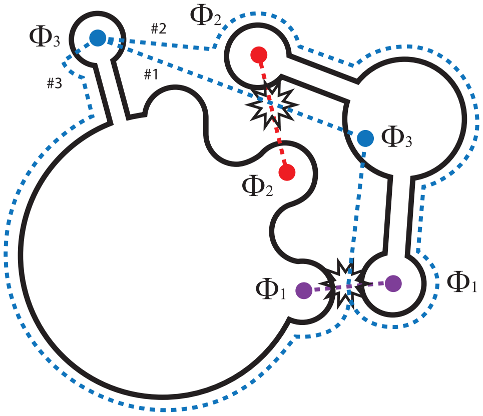


Figure 5.18: Example of a routing failure. In this case, it is not possible to find a route that connects port pair Φ_3 that doesn't cross the routes for at least one of port pairs Φ_1 and Φ_2.

Once the probe is generated, a Channel component is instantiated for the probe and merged (Lines 37) into c using standard Bézier intersection and merging algorithms [91] where it will become part of the outline for the next probes to avoid. If the probe cannot be deflected around a routed channel, the routing fails and what was successfully routed is returned along with an error (Fig. 5.18). The iterating values of i and j are incremented (Lines 38-39) and the next pairs are evaluate If the ports are all successfully routed, then Chs is returned without error (Fig. 5.19, Lines 40).

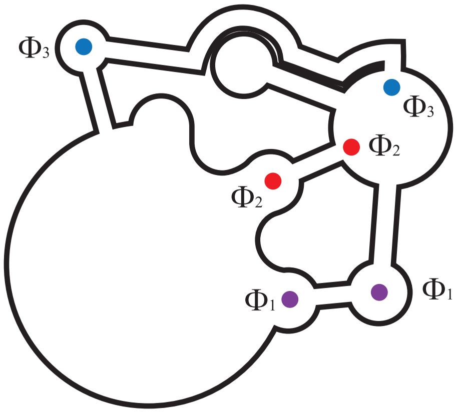


Figure 5.19: A successfully routed layout. The newly introduced component and routed channels are integrated into the device. The algorithm is ready to place-and-route the next component.

5.2 Experimental Results

5.2.1 Methodology

As the microfluidic space is non-discrete, the DICE[16] algorithm was chosen as an appropriate benchmark to compare the Radar approach. Although DICE is built to a grid, the units are based on the physical dimensions of the components with a user-specified buffer between them. Components are placed down and to the right of placed components yielding a diagonal placement that allows for routing to be performed in a mostly-linear fashion. Once routed, the layout is then rotated 45° to reduce it's footprint for fabrication

Testing harnesses were built to generate tests for area utilization. Three test cases were developed for comparing DiCE (Fig. 5.20a,c,e) and RADAR (Fig. 5.20b,d,f): a "chain" of components (Fig. 5.20a,b) that are directly connected one time from source to sink for each component, an "orbital" approach (Fig. 5.23c,d) where a single source is connected to any number of sinks surrounding it, and a "tree" (Fig. 5.24e,f) where each component is connected to 2 sinks. All tests utilize a directed acyclic graph for maintaining connections thus protecting against loops, however both DICE and Radar can handle cycles without indefinite loops. Several sets of component counts were chosen as initial tests to illustrate potential trends due to layout growth.

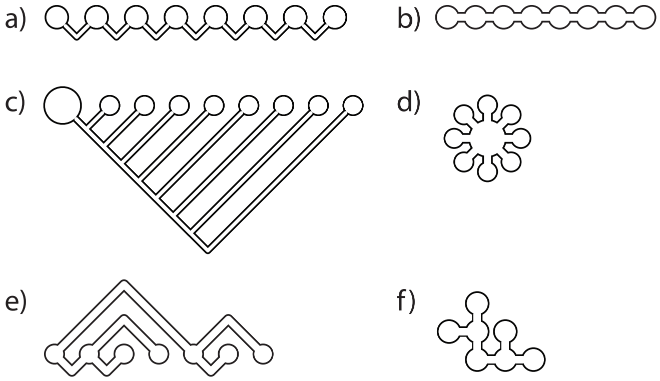


Figure 5.20: Three test cases were developed to evaluate the quality of layouts produced by DICE [16] (left column) vs. RADAR (right column): a,b) depict the "Chain" test that connects each source to the next sink sequentially; c,d) shows the "Orbit" layout where several sinks are connected to a central source; and e,f) the "Tree" structure where each component is connected to two sinks.

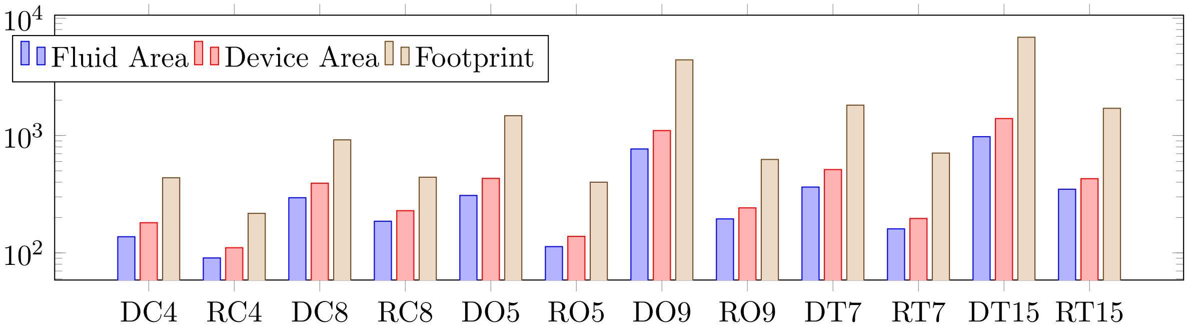


Figure 5.21: Area utilization in terms of fluid space, device occupancy, and total area required for fabrication (in mm^2).

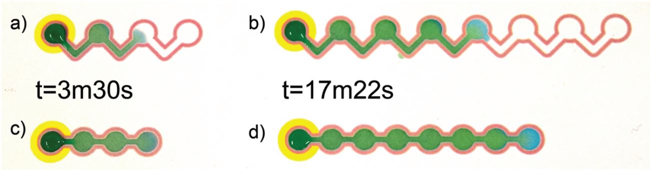


Figure 5.22: The "Chain" layout consisting of a number of components connected in a series. DiCE layouts of a) 4 components and b) 8 components, RADAR versions are c) 4 and d) 8 components. Times listed refer to completion of devices for the RADAR versions and the state of the DiCE versions. Table 5.3 lists the completion and/or failure times for the DiCE versions.

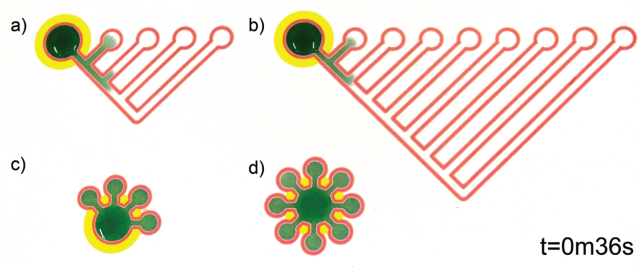


Figure 5.23: The “Orbit” layout consisting of a single source feeding multiple sinks shows the importance of compact layouts. Layouts a-b) use DiCE, c-d) use Radar.

Each layout was then output onto LabNerd® paper stock, reheated to sublimate the wax ink into the paper, and a PCR backing tape applied to isolate the layout from the work surface. Filtered water with several drops of green food coloring was pipetted into the source region of each layout. Initially 40 μL of fluid was delivered with additional fluid delivered in 20 μL steps if the device showed signs of drying out until the fluid movement reached all sinks in the layout, or the fluid ceased to travel any further due to the leading edge drying out forming a barrier to additional flow.

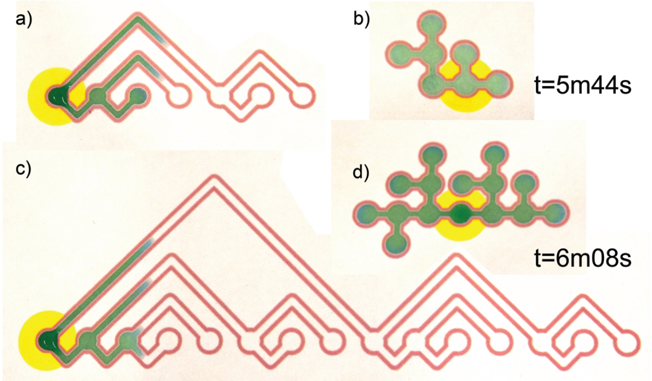


Figure 5.24: The “Tree” layout consisting of each node feeding 2 nodes. a,c) DiCE, b,d) RADAR

Table 5.2: Runtimes for each DiCE and RADAR for each layout test, along with the number of components placed and routed. Δ indicates the difference in runtime. RADAR was significantly slower in all cases; this is to be expected as RADAR is exhaustive while DiCE is a heuristic.

Test	Algorithm	#	Time	Δ
Chain	DiCE	4	0:00:04	
	RADAR	4	0:05:48	**0:05:44**
	DiCE	8	0:00:10	
	RADAR	8	1:25:54	**1:25:44**
Orbit	DiCE	5	0:00:05	
	RADAR	5	0:14:55	**0:14:50**
	DiCE	9	0:00:11	
	RADAR	9	1:16:11	**1:16:00**
Tree	DiCE	7	0:00:09	
	RADAR	7	0:39:39	**0:39:30**
	DiCE	15	0:00:27	
	RADAR	15	1:46:49	**1:46:22**

5.2.2 Algorithm Results

Results are plotted such that the vertical axis plots area usage measured in mm^2 on a logarithmic scale (Fig. 5.21). The horizontal axis shows the individual test results with the first letter being the algorithm (D)ICE or (R)adar; the second letter the test performed: (C)hain, (O)rbit, or (T)ree; and lastly, the number of components placed and routed. In each test and for nearly every metric, Radar outperformed DICE in area utilization. In terms of fluid area, Radar would outperform DICE due to being able to more compactly place components thereby shortening the channel and therefore the fluid travel. The chain test allowed for a more level playing field due to reducing the amount of channel deflection.

Notable however, both DiCE and Radar initially failed routing of all 15 components during the tree test which indicated a weakness in their approaches when the "greedy" criteria of lowest $snugnessfactor$ produces a layout too compact to route past the second level of the tree. It was determined that presenting the tree netlist in a breadth-first manner resulted in the algorithms' inability to place and route and when the netlists were traversed in a depth-first manner, both algorithms were able to successfully place and route the netlists.

RADAR is intended to explore the paper microfluidic device layout space as completely as possible. High execution times are expected, and this will inevitably limit scalability. As shown in Table 5.2, RADAR runs several orders of magnitude slower than DiCE (a heuristic), due to the ever-expanding set of critical points that it enumerates. RADAR examines $\Theta(m \times n)$ critical points, per pass, as candidate locations for routing. While the routing phase can terminate in $\Omega(1)$ time if the first candidate is routed successfully, it is also possible that routing may fail for all candidates. Consequently, runtimes can become extremely large, for example, as RADAR took nearly 2 hours to complete route the 15-component "Tree" benchmark. Even the smaller case, a 4-component "Chain," required nearly six minutes to complete. Future work may examines strategies to reduce the runtime of RADAR, including techniques that limit the portion of the search space explored, as well as a parallel implementations of key bottlenecks.

5.2.3 Physical Device Performance Results

The physical devices output from generated layouts were tested for real-world performance by delivering fluid to each device's source region and monitored for either failure to complete or time to completion. Although DiCE did successfully generate placed and routed layouts, not all devices were able to run to completion. Devices featuring long channels and distant sinks would fail due to the leading edge of the fluid drying out and forming a barrier to any further fluid travel – even when additional fluid delivered to the device. As noted in Table 5.3, only two versions of the DiCE layouts ran to completion even though additional fluid (amounts listed) was dispensed. By comparison each Radar device was not only able to run to completion with only 20 μL of additional fluid required for the eight component chain and tree versions (Figs. 5.22 and 5.24 respectively). As indicated by the Δ column, devices laid out by RADAR devices have factors of improvement in the range of $3.5 - 113\times$ over those laid out by DiCE, while successfully running to completion in all cases.

Table 5.3: Experimental results listing the which test, algorithm used, the number of components placed and routed, the volume of fluid delivered, and the time for the device to complete or the time of failure (in bold) due to the leading edge of the fluid drying out and blocking any further flow. Δ lists in red the additional time required for the devices laid out by DiCE to run to completion, compared to the devices laid out by RADAR.

Test	Algorithm	#	μL	Time (h:m:s)	Δ
Chain	DiCE	4	40.0	00:12:16	8:46
	RADAR	4	40.0	00:03:30	**3.5x**
	DiCE	8	120.0	**01:11:40**	54:18
	RADAR	8	60.0	00:17:22	**4x**
Orbit	DiCE	5	100.0	00:43:48	43:12
	RADAR	5	40.0	00:00:36	**73x**
	DiCE	9	200.0	**00:52:38**	52:10
	RADAR	9	40.0	00:00:28	**113x**
Tree	DiCE	7	120.0	**00:45:48**	40:04
	RADAR	7	40.0	00:0544	**8x**
	DiCE	15	140.0	**00:45:30**	39:22
	RADAR	15	60.0	00:06:08	**7x**

Part III

Device Experimentation

Chapter 6

Using Printer Ink Color to Control Fluid Behavior

Many paper microfluidic devices are fabricated using the Xerox ColorQube line of commercial wax-based color printers; the wax ink serves as a hydrophobic barrier to fluid flow. These printers are capable of depositing four different colors of ink, cyan (C), magenta (M), yellow (Y), and black (K), plus 11 combinations of these colors (CM, CY, CK, MY, MK, YK, CMY, CMK, CYK, MYK, and CMYK), although most researchers use only black ink to print paper microfluidic devices. While developing the software framework it was unexpectedly observed that different colors of wax ink behave differently in paper microfluidics. It was found that among the single colors of ink, black ink actually had the most barrier failures, and magenta ink had the fewest barrier failures. In addition, some combinations of colors performed even better than magenta: the combinations CY, MK, YK, CMY, CYK and MYK had no barrier failures in this study. It was also found that the printer delivers significantly different amounts of ink to the paper for the different color combinations, and in general, the color combinations that formed the strongest barriers to fluid flow were the ones that had the most ink delivered to the paper. This suggests that by simply weighing paper samples printed with all 15 combinations of colors, one can easily find the color combinations most likely to form a strong barrier for a given printer. Finally, to show that deliberate choices of ink colors can actually be used to create new functions in paper microfluidics, a new color-based “antifuse” structure was designed and tested that protects paper microfluidic devices from a typical operator error (addition of too much fluid to the device). The results could provide a set of color choice guidelines that designers can use to control the behavior of their paper microfluidics.

One of the most common techniques for fabricating paper microfluidic devices utilizes the Xerox ColorQube line of color office printers. Instead of liquid ink or powdered toner, ColorQube printers (like the one shown in Figure 6.1) contain solid blocks of wax-based ink, which the printer melts and deposits on the paper. Chandler *et al.* [12] showed that since this wax ink is hydrophobic, it can be used to make channel walls and define fluid paths in paper microfluidics.

Recently, as part of a project to develop a computer-aided design (CAD) framework for use with paper microfluidics,[86] it was unexpectedly observed that the different colors of ColorQube wax ink behave differently in paper microfluidics. This observation may have gone unnoticed by previous researchers because even though ColorQube printers can print wax of four different colors (cyan, magenta, yellow, and black) plus all combinations of these colors, most researchers use only the black ink when printing their paper microfluidic devices. In one notable exception, Taudte *et al.* [93] used a few different ink colors in their paper microfluidic devices for detecting explosive materials, and concluded that magenta ink served as a more robust barrier to fluid flow than black ink. However, as far as is known, no one has systematically explored all 15 possible combinations of cyan, magenta, yellow, and black that can be delivered by ColorQube printers, or identified color combinations with behaviors that make them suitable for functions other than fluid barriers, or offered a possible explanation for the different behaviors of different ink colors.

In this chapter, the effect of ColorQube wax ink color on paper microfluidic device behavior is systematically explored. The study required the creation of a large number of different paper microfluidic device designs, and the paper microfluidics CAD framework was leveraged to automate and expedite the generation of these device designs.[86] These findings provide a set of rules on color selection that researchers can use to easily control the behavior of their paper microfluidics. An explanation for why different colors behave differently is provided that demonstrate that a series of simple mass measurements can be used to determine which color combinations will likely provide the strongest barriers to fluid flow for a given printer. Finally, a useful new paper microfluidic component—a paper microfluidic "antifuse"—that uses different ink colors to create the desired behavior is demonstrated.

6.1 Materials and Methods

6.1.1 Overview of color printing with the CMYK model

Xerox ColorQube printers use solid blocks of wax-based ink in four different colors: cyan (C), magenta (M), yellow (Y), and black (K). This means that the printers use the *CMYK model*, a subtractive color model that uses secondary colors to reproduce the full range of colors.[58] To reproduce primary colors, CMYK printers generally combine two of the secondary colors (*e.g.*, magenta and yellow are combined to form red; see Figure 6.1 for examples). The black ink is used to produce darker shades of the colors in the CMYK model but is not generally used by itself where a full black color is desired when printing. Rather, a percentage of cyan is also combined with black to create a more opaque black color. Finally, a combination of all four inks known as "registration" black is used for aligning inks when using presses that print each color separately. In summary, 15 different combinations of ink colors can be generated by these printers: C, M, Y, K, CM, CY, CK, MY, MK, YK, CMY, CMK, CYK, MYK, and CMYK. By controlling the color of a device feature in a graphical design program, the user can easily create paper microfluidic device features in any of these color combinations.

6.1.2 Automated design of paper microfluidic devices

Instead of designing the paper microfluidic devices by hand using graphic design software, in this work the previously discussed computer-aided design (CAD) framework [86] was used to automatically generate the necessary device designs. The framework can accelerate the development of new paper microfluidic devices and reduce the material required to

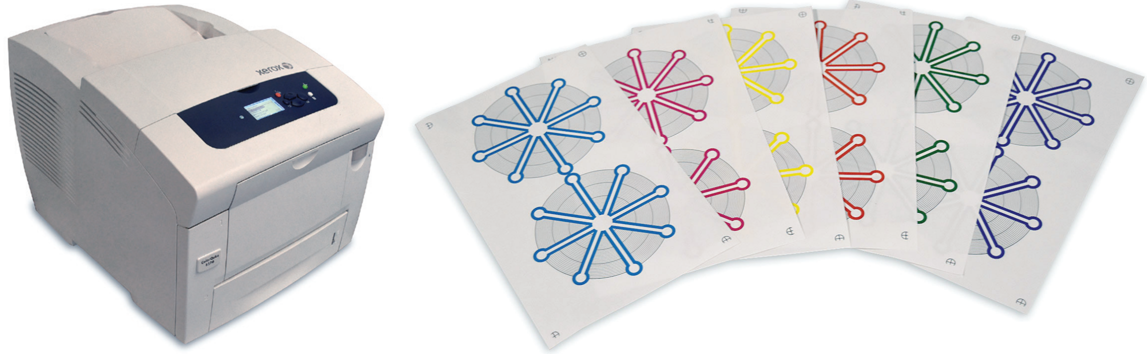


Figure 6.1: **(Left)** Xerox ColorQube wax-based color printers (like this ColorQube 8570 model) are routinely used for creating paper microfluidic devices. **(Right)** Paper microfluidic test devices for six of the 15 different wax ink color combinations that were tested (from left): cyan, magenta, yellow, magenta + yellow, cyan + yellow, and cyan + magenta.

make these devices. Developers can use the framework to prototype, dynamically generate, and test new designs. These designs may account for the effects of different environmental conditions, physical substrates, and fluid conditions, providing the designer with a greater understanding of how these physical factors influence the behavior of a microfluidic assay.

The framework contains a library of components for developing devices, as well as pre-defined devices that are parameterized to allow for quick alteration of specifications. In this work, the library's "bullseye" calibration device shown in Figure 6.1 was used. This design provides the user with a radial array of channels that connect a central "source" to individual "sinks" around the edges of the device as well as markers alongside the channels to facilitate measurements of fluid travel. The user specifies the various parameters of the desired "bullseye" device, including the number, width, and angle of the channels, the diameters of the sources and sinks, and the widths and colors of the wax ink channel barriers. The software framework then generated a graphics file ready to print on the color printer. In this manner, all 48 different "bullseye" device designs used in this study were easily generated in less than a second per device. Over the course of this study, the “antifuse” design (described below) was added as a new component in the library to facilitate the inclusion of this component in future paper microfluidics.

6.1.3 Fabricating paper microfluidic devices

Currently, Xerox ColorQube printers are the only wax-ink-based printers on the market. Xerox has sold 12 different models of ColorQube printers, and the exact compositions of the inks used by the printers are trade secrets. However, close inspection of the inks’ Material Safety Data Sheets (MSDSs)[112, 110, 111] reveals that the inks used by all 12 of the ColorQube printers share the same basic chemical composition (by weight):

- 50–60% paraffin wax
- 10–20% resin
- 0–10% blue dye
- 0–10% red dye
- 0–10% yellow dye
- 0–10% black dye

In other words, the inks used by all 12 models of ColorQube printers are mostly paraffin wax and resin, with the remainder consisting of one or more dyes that are specific to each ink color. Additionally, the other ink data provided by the MSDSs (hazards, melting point, flash point, solubility, specific gravity, chemical regulatory data, etc.) are all identical across the different ColorCube models. This suggests that the composition of each ink color is very similar (and possibly identical) across the 12 different ColorQube printers, although the exact compositions are trade secrets.

In this work,a Xerox ColorQube 8570 printer (Figure 6.1) was connected via USB to a PC loaded with the CAD framework running on the Linux operating system. The generated device designs were printed on “Lab Nerd 101” fast qualitative filter paper (Amazon.com). Since this filter paper comes from the manufacturer as 24 cm diameter circles but the ColorQube printer requires rectangular paper for printing, each sheet was cut down to a 200 mm × 123 mm rectangle for manually feeding into the printer. This provided enough space for two separate paper microfluidic test devices per sheet. After printing, each pair of devices was heated over a laboratory hotplate to re-melt the wax ink and form a hydrophobic barrier across the full thickness of the paper. The hotplate was pre-heated to 166 °C (as measured using an infrared thermometer), then each printed paper sample was held 5–7 cm above the heating element (with the printed side facing upward) until melting was observed in the wax ink (ink color faded and the edges of the ink features blurred). At this point, the Bullseye Calibration Devices were ready to use in experiments. The “anti-fuse” test paper microfluidic devices had self-adhesive PCR plate sealing film (Microseal B; Bio-Rad, Hercules, CA) applied to the backside (non-ink side) of the paper before use; this tape keeps fluid from flowing through the backside of the paper.

6.1.4 Testing paper microfluidic devices

The “bullseye” paper microfluidic devices (shown in Figure 6.1) were tested using a custom rig built out of finished lumber (Figure 6.2). Each paper device was clamped into the leveled rig and suspended from the edges to eliminate any contact between the work surface and the backside of the paper device. The devices were tested using deionized water containing a small amount of food coloring (3 drops per 50 mL) delivered to the paper using either a pipette (for the black ink color studies) or a mounted low-flow-rate nozzle (Antelco; Longwood, FL; for the color ink studies, shown in Figure 6.2; the nozzles delivered 300 μL of fluid to each paper microfluidic device at a rate of 200 μL per second). Along with photographing each device during fluid delivery, the time was also recorded when fluid reached the first sink and the time when fluid reached the eighth and final sink.

The "antifuse" paper microfluidic devices were tested on a bench top. A pipette was used to repeatedly deliver 50 μL volumes of fluid to the source reservoir of the device until flow across a wax ink barrier was observed.

6.1.5 Analysis of the amount of wax ink deliverer by printer

For studies of the amount of each ColorQube ink deposited on the paper by the printer, a laboratory analytical balance was used to weigh paper samples. Ink mass was determined by first measuring the pre-print mass of a paper substrate along with a binder clip used to hold the paper in a cylindrical shape to accommodate the interior dimensions of the balance enclosure. The mass of the clip was then subtracted from subsequent measurements. A 17 cm by 10 cm solid rectangle of ink was then printed on the paper. After printing, the paper was weighed again and the two mass measurements were subtracted to determine the mass of the ink deposited by the printer. Dividing this mass by 170 cm^2 (the area of the ink rectangle) yields the mass of ink deposited per unit area. This measurement

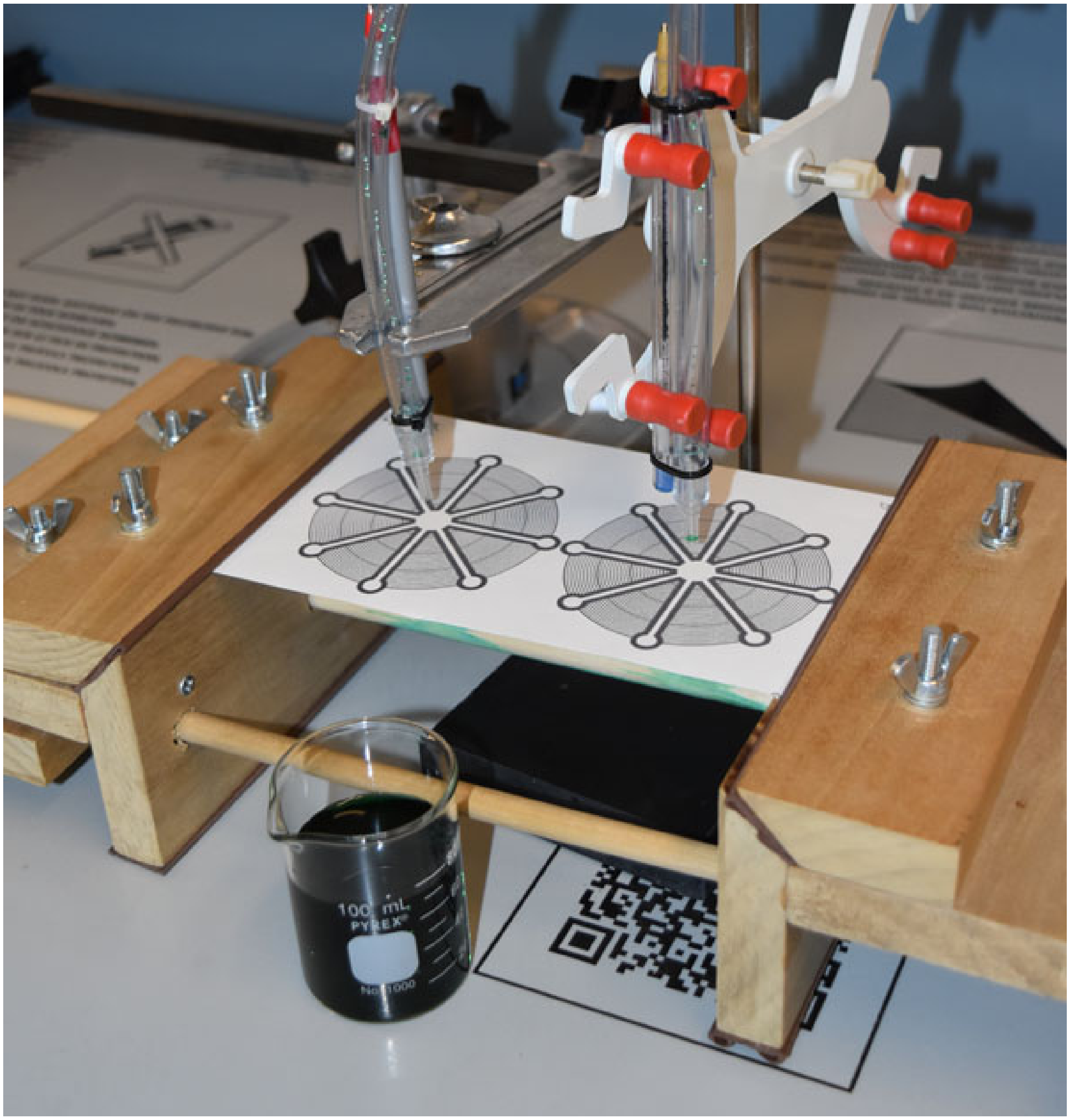


Figure 6.2: Custom-built test rig for dispensing known volumes of fluid at known flow rates to paper microfluidic devices.

was repeated and recorded for all 15 solid colors (single colors and combinations) that can be generated by the printer.

6.2 Results and discussion

6.2.1 Barrier width testing with black ink

Preliminary experiments were performed to determine the necessary ink barrier width for reliable fluid containment using black ColorQube ink (which is the usual ink color used in paper microfluidics). The “bullseye” device was used from the framework library, configured with 8 radial channels, 2 mm wide channels, 6 mm diameter sinks, 12 mm diameter sources, and 86 mm overall diameter. Ink barrier widths were varied from 1.0 to 2.0 mm (Fig. 6.3), and 16 devices were tested for each width (a total of 48 different tests). A pipette was used to deliver 300 μL of dyed water to the central source of each “bullseye” pattern, and the fluid was allowed to travel outward to the eight sinks on the perimeter, stopping when the fluid had reached one or more of the sinks. Failures were categorized as either “flow through barrier” (when fluid wicked through the ink-impregnated barrier) or “flow over barrier” (when fluid flowed across the top of the ink barrier).

Results from the black barrier width study are summarized in Table 6.1 row K, and typical results are shown in Figure 6.3. The thinnest black ink barriers (1.0 mm thick) consistently failed by having the fluid flow over the thin barrier. At moderate barrier thicknesses (1.5 mm), failures occurred most often due to flow through the barrier. Finally, devices with the largest barrier thicknesses (2.0 mm) exhibited the fewest total failures.

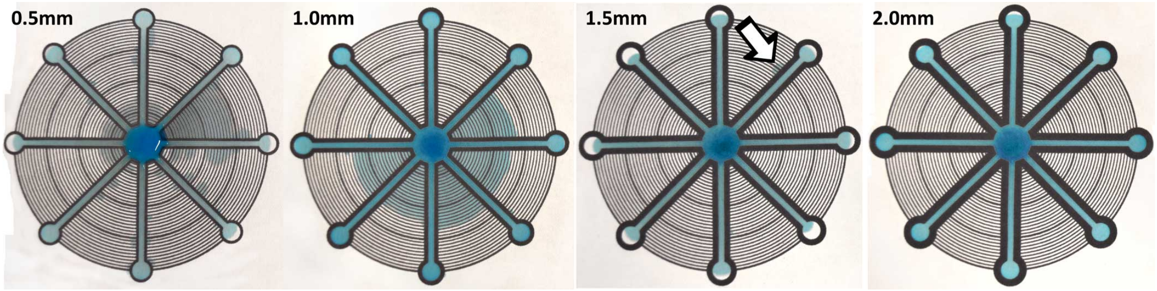


Figure 6.3: Sample results from experiments to determine reliable barrier thickness for black ColorQube ink using the software framework's "bullseye" test device. Consistent barrier failure occurred at black barrier widths of 1.0 mm and smaller. The 1.5 mm barrier did a much better job of containing fluid, but some failures were still observed (as indicated by arrow). Ultimately, a 2.0 mm barrier width was found to have the fewest barrier failures.

Fluid took from 144 s to 488 s to reach the first of the eight sinks, with a median time of 392 s. Fluid took from 157 s to 560 s to reach the last sink, with a median time of 443 s.

6.2.2 Barrier color testing with individual cyan, magenta, and yellow inks

The barrier tests shown in Figure 6.3 were repeated, but this time using the other individual ink colors provided by ColorQube printers (cyan, magenta, and yellow) instead of black. The results summarized in Table 6.1 (lines C, M, and Y) reveal significant differences in how the different individual ink colors perform as barriers to fluid flow. Interestingly, black ink (K), which is used most often in wax-printed paper microfluidic devices, actually performed the worst of all the single-color barriers in every barrier width, with 21 failures out of 48 total tests or a 44% failure rate. Magenta ink (M) performed the best, with only three failures observed in the thinnest barriers and no failures in the other barrier thicknesses (a combined failure rate of only 6%). These results are consistent with the observations of Taudte *et al.* [93] and suggest that **for maximum resistance to unwanted fluid flow**

Ink colors	Barrier width	Flow through barrier	Flow over barrier	Total Total failures	Ink colors	Barrier Barrier width	Flow through barrier	Flow over barrier	Total Total failures
C	1.0 mm	**5/16**	0/16	**5/16**	MY	1.0 mm	0/16	**5/16**	**5/16**
	1.5 mm	**1/16**	0/16	**1/16**		1.5 mm	0/16	0/16	0/16
	2.0 mm	**1/16**	0/16	**1/16**		2.0 mm	0/16	0/16	0/16
M	1.0 mm	0/16	**3/16**	**3/16**	YK	1.0 mm	0/16	0/16	0/16
	1.5 mm	0/16	0/16	0/16		1.5 mm	0/16	0/16	0/16
	2.0 mm	0/16	0/16	0/16		2.0 mm	0/16	0/16	0/16
Y	1.0 mm	**6/16**	**5/16**	**11/16**	CMK	1.0 mm	**16/16**	0/16	**16/16**
	1.5 mm	**3/16**	**2/16**	**5/16**		1.5 mm	**16/16**	0/16	**16/16**
	2.0 mm	0/16	**5/16**	**5/16**		2.0 mm	**15/16**	0/16	**15/16**
K	1.0 mm	0/16	**6/16**	**6/16**	CMY	1.0 mm	0/16	0/16	0/16
	1.5 mm	**11/16**	0/16	**11/16**		1.5 mm	0/16	0/16	0/16
	2.0 mm	**4/16**	0/16	**4/16**		2.0 mm	0/16	0/16	0/1
CK	1.0 mm	0/16	**2/16**	**2/16**	CYK	1.0 mm	0/16	0/16	0/16
	1.5 mm	0/16	0/16	0/16		1.5 mm	0/16	0/16	0/16
	2.0 mm	0/16	0/16	0/16		2.0 mm	0/16	0/16	0/16
CM	1.0 mm	**2/16**	**2/16**	**4/16**	MYK	1.0 mm	0/16	0/16	0/16
	1.5 mm	0/16	0/16	0/16		1.5 mm	0/16	0/16	0/16
	2.0 mm	0/16	0/16	0/16		2.0 mm	0/16	0/16	0/16
CY	1.0 mm	0/16	0/16	0/16	CMYK	1.0 mm	**5/16**	0/16	**5/16**
	1.5 mm	0/16	0/16	0/16		1.5 mm	**8/16**	0/16	**8/16**
	2.0 mm	0/16	0/16	0/16		2.0 mm	**1/16**	0/16	**1/16**
MK	1.0 mm	0/16	0/16	0/16					
	1.5 mm	0/16	0/16	0/16					
	2.0 mm	0/16	0/16	0/16					

Table 6.1: Occurrences of barrier failures using single ink colors black (K), cyan (C), magenta (M), and yellow (Y), and all combinations of these colors, at various barrier widths. "Flow through barrier" failures occurred when fluid was observed to wick through the ink-impregnated barrier and out of the channel region, and "flow over barrier" failures occurred when fluid was observed to flow across the top of the ink barrier and out of the channel region. Among single-color inks, black ink (K) performed the worst (failing in 21/48 or 44% of tests), and magenta ink (M) performed the best (failing in only 3/48 or 6% of tests). Among combinations of ink colors, the combinations CY, MK, YK, CMY, CYK, and MYK had no flow failures at any of the barrier widths tested.

through or across the ColorQube wax ink barriers, when choosing one of the printer's four individual ink colors, creators of microfluidic devices should use magenta ink instead of black.

An unusual behavior in test devices with yellow barriers was observed. Table 6.1 shows that while no "flow over barrier" failures were observed in the thicker black, cyan, and magenta barriers, several "flow over barrier" failures were observed in the thicker yellow barriers. Figure 6.4 shows typical yellow "bullseye" devices both before and after a "flow over barrier" failure; the blue fluid is flowing across two millimeters of yellow wax ink barrier. These observations suggest that the surface of paper impregnated with yellow wax ink may be less hydrophobic than the other colors. These observations also suggest that the distinctive failure mode of yellow ink barriers could be used by designers to impart certain behaviors to paper microfluidic devices, an idea which was then tested as well.

6.2.3 Barrier color testing with combinations of ink colors

As noted above, ColorQube printers are capable of printing colors other than cyan, magenta, yellow, and black by simultaneously delivering *combinations* of two or more colors to the paper. It was hypothesized that these combinations may result in more wax ink being delivered to the paper (therefore creating a more effective barrier to fluid flow) compared to the single ink colors. To test this hypothesis, the above tests were repeated using devices made with all possible combinations of cyan, magenta, yellow, and black (CM, CY, CK, MY, MK, YK, CMY, CMK, CYK, MYK, and CMYK). In each combination, each color is specified at 100% density, meaning that the printer is instructed to deliver as much of each

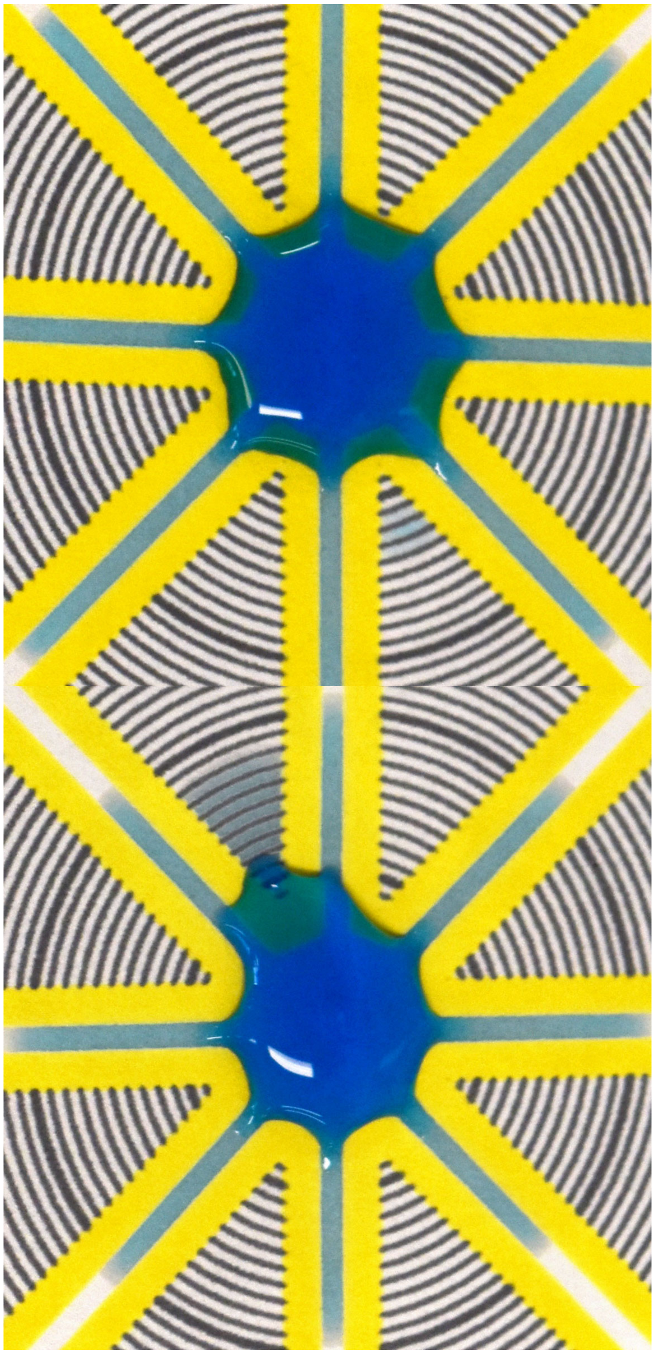


Figure 6.4: Photographs of two test devices with 2.0 mm wide yellow ColorQube ink barriers, one immediately before a “flow over barrier” failure (top) and one immediately after (bottom). Both photos were taken 2 minutes and 16 seconds into the experiment.

color as possible. This means that all color combinations that include black will appear black when printed, but will still contain the additional ink colors.

The results in Table 6.1 show that most devices containing multiple color barriers fared better than the single-color barriers. Of the 11 possible color combinations, 6 of them (CY, MK, YK, CMY, CYK, and MYK) had no flow failures at any of the barrier widths tested. **These color combinations are excellent choices for robust barriers, especially at thinner barrier widths.** They outperformed all of the single-color barriers. An additional three combinations (CK, CM, and MY) failed only for the thinnest barrier, a performance on par with magenta (the best single-color barrier).

However, two combinations of wax colors performed far worse than the other combinations. Specifically, all but one of the 48 test devices printed using CMK barriers suffered from "flow through barrier" failures (a staggering 98% failure rate) and 14 of the 48 test devices printed using CMYK barriers suffered the same failure (a 29% failure rate). It was hypothesized that these failures might be attributed to variation in the amount of ink the printer deposits in the paper when printing these color combinations, a hypothesis that was tested in the next section.

Measuring the mass of ink deposited on the paper

To gain insights into why different ink color combinations behave dramatically differently in paper microfluidic devices,the mass of ink being deposited by the printer was measured for each single color and color combination, both before and after heating. The results (Table 6.2 and Figure 6.5A) reveal significant variation in the total amount of ink

Colors	Pre-heating ink mass (mg cm^{-2})	Post-heating ink mass (mg cm^{-2})	Ink mass loss (mg cm^{-2})
K	1.91	1.62	0.29
M	1.36	1.05	0.31
C	1.13	0.79	0.34
Y	0.96	0.68	0.29
CK	1.86	1.54	0.33
CY	1.86	1.55	0.31
YK	1.81	1.46	0.35
MY	1.68	1.35	0.34
CM	1.59	1.28	0.31
MK	1.22	0.92	0.30
CMY	1.90	1.63	0.26
MYK	1.26	0.90	0.36
CYK	1.09	0.85	0.25
CMK	0.89	0.62	0.28
CMYK	0.90	0.65	0.26

Table 6.2: Masses of ink deposited on the paper, for each single color (C, M, Y, and K) and combinations of colors, before and after heating the paper, as well as the mass lost due to heating. The results show a wide range in the amount of ink deposited for the different colors (ranging from 0.89 mg/cm^2 for CMK to 1.91 mg/cm^2 for K), though all colors lost roughly the same amount of mass during heating (about 0.3 mg/cm^2).

delivered by the printer for different color combinations. The color combination with the largest mass of ink delivered to the paper (K; 1.91 mg/cm^2), has over twice the mass of the combination with the smallest mass of ink delivered (CMK; 0.89 mg/cm^2). Also, it was observed that a significant amount of this ink mass is lost when the paper is heated. A plot of pre-heating ink mass vs. post-heating ink mass (Figure 6.5A) shows that each color combination lost roughly the same amount of mass during heating (about 0.3 mg/cm^2). This suggests that for each color combination, the printed ink contains about 0.3 mg/cm^2 of volatile components that are lost during heating, plus a variable amount of less-volatile pigments and waxes that remain on the paper after heating.

Since paper microfluidic devices with wax inks must be heat-treated before use, the remaining analysis was focused on post-heat-treatment ink masses. Figure 6.5B plots the rate of barrier failures vs. post-heating ink mass for all 15 ink color combinations. The results show that (with one exception) **the color combinations with largest mass of ink delivered to the paper have the lowest rates of barrier failure**. The one exception to this trend is black ink (K), which had poor barrier performance despite having the second-highest post-heating ink mass. This suggests that the black ink contains a relatively large amount of pigment (necessary for a solid black color), and this pigment reduces the color's effectiveness as a barrier to fluid flow, perhaps by reducing the hydrophobicity of the ink, or occupying space that would otherwise be filled by hydrophobic wax.

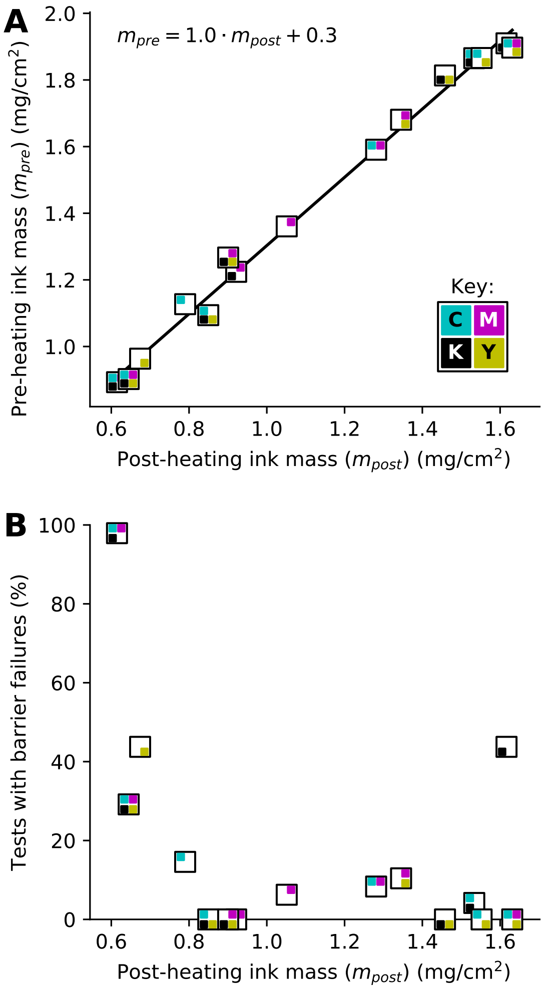


Figure 6.5: **(A)** Plot of the mass of ink delivered by the printer for all 15 combinations of ink colors, before (y-axis) and after (x-axis) heating the paper samples to re-melt the ink and form a hydrophobic barrier. While the total mass of ink delivered by the printer per square centimeter of paper area varies significantly with color combination, all samples lost roughly the same amount of weight during heating (about 0.3 mg/cm^2). **(B)** Plotting the fraction of tests with barrier failures from Table 6.1 vs. the post-heating mass of ink delivered by the printer for the same combinations of ink. In general, color combinations with more ink delivered to the paper resulted in fewer barrier failures. The main exception to this trend was black (K), which suffers from significant barrier failures despite having one of the largest masses of ink delivered to the paper.

6.2.4 The paper microfluidic "antifuse"

Finally, to show that deliberate choices of ColorQube ink colors can actually be used to create new functions in paper microfluidics, these findings were to design and test a paper microfluidic "antifuse," an ink structure that deals with excess fluid in a controlled manner. In electronics, an antifuse is a component that functions as the opposite of a fuse; it normally provides a high resistance to electric current, but becomes low resistance when the voltage across it exceeds a certain level. The paper microfluidic antifuse functions similarly by providing an initial hydrophobic barrier that resists fluid flow until a certain volume of fluid is delivered to the component, at which point the antifuse becomes a low resistance path that carries excess fluid to a desired location.

The main feature in the antifuse test device shown in Figure 6.6A is a horizontal channel for fluid flow; fluid is added to the circular pad on the left (marked "source") and flows by wicking through the channel and to the circular pad on the right (marked "sink"). In a real application, this horizontal channel might contain immobilized reagents that serve as readout lines in a lateral flow assay. Most of the hydrophobic barriers around this channel are magenta, chosen because magenta offers the greatest resistance to fluid flow of all the single ink colors. However, yellow ink was used for the barrier around the source pad which receives added fluid. It was hypothesized that by using yellow wax ink for this antifuse barrier, it could be possible to take advantage of the tendency of yellow ink to exhibit flow-over failures like the ones shown in Figure 6.4. Specifically, if excess fluid is added to the source pad, the fluid would flow over the yellow antifuse barrier and be contained in the large "moat" surrounded by the larger circular magenta barrier, instead of overflowing a

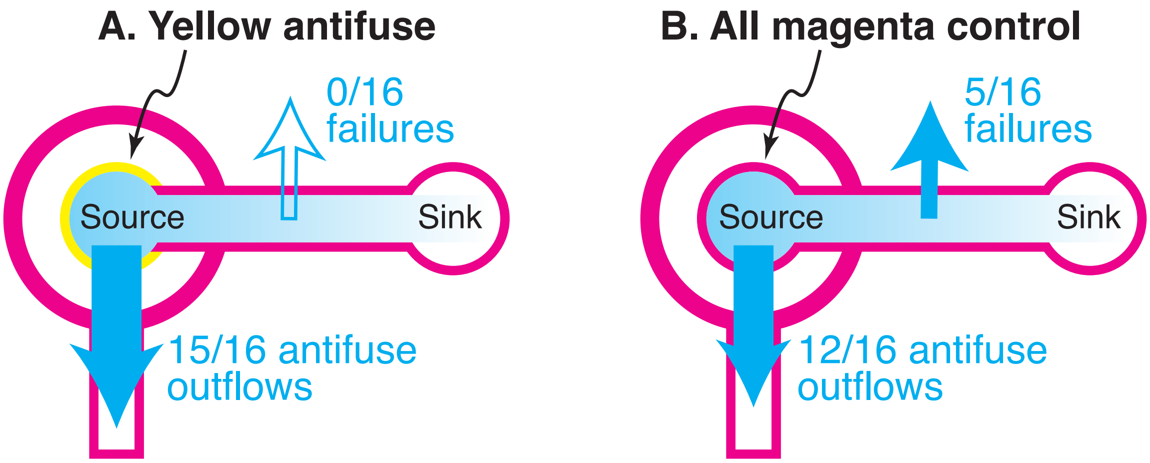


Figure 6.6: Summary of results from 32 tests of an ink-color-based "antifuse" for paper microfluidics. **(A)** When an operator adds excess fluid to the left-side circular pad marked "source," the excess fluid flows over the yellow antifuse barrier and is trapped in the large circular "moat" in 15 of 16 tests, thereby protecting the rest of the device (where no failures were observed in all 16 tests). **(B)** In contrast, when a magenta antifuse barrier is used, excess fluid overflows into the "moat" less frequently (in 12 of 16 tests), meaning more fluid flows toward toward the "sink" pad and causes more device failures (5 of 16 tests). These results show that a yellow antifuse barrier can serve to protect paper microfluidic devices from a common operator error (the addition of excess fluid).

barrier elsewhere in the device and likely invalidating the assay. In this manner, the yellow antifuse structure would enable the paper microfluidic channel to automatically recover from a typical operator error, the addition of too much fluid to the paper microfluidic device.

To test the antifuse structure, the software framework was used to create two different versions of the antifuse test device: one containing a yellow antifuse channel and expected to function as described above (Figure 6.6A), and one containing a magenta antifuse channel and intended to serve as an all-magenta "control" expected to fail at unpredictable locations (Figure 6.6B). Sixteen of each of these designs were then fabricated and tested each by delivering water via pipette in $20 - 50$ μL increments to the source pads while watching for either flow over the antifuse barrier or failures elsewhere in the device.

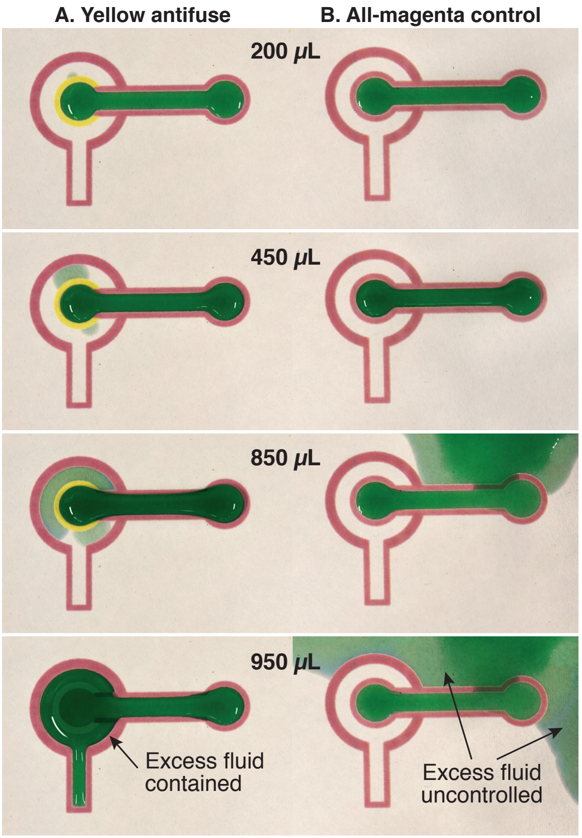


Figure 6.7: Photographs of two typical antifuse test devices while adding fluid to the left-side circular source pad. As more fluid is added to the device with a yellow antifuse barrier **(A)**, the excess fluid flows over the yellow antifuse and into the surrounding "moat," protecting the rest of the device from barrier failures. In contrast, when excess fluid is added to the device with a magenta antifuse **(B)**, the barrier fails catastrophically in an unpredictable location, likely invalidating the assay. These results show that yellow antifuse structures can protect a paper microfluidic device from some types of failures.

For the 16 tests with a yellow antifuse (Fig. 6.6A), the excess water flowed over the antifuse barrier in 15 of the 16 experiments; no failures occurred elsewhere in the device. This means that the yellow antifuse functioned as intended, routing excess fluid into a specified overflow area and protecting the rest of the device from barrier failures. Photographs of a typical yellow antifuse test device in operation are shown in Figure 6.7A; as the amount of fluid added to the Source pad increases, the flow across the yellow antifuse barrier also increases, protecting the rest of the device from barrier failures.

In contrast, for the 16 tests with a magenta antifuse, flow over the antifuse barrier was less common (occurring in 12 of the 16 experiments), meaning that more of the excess fluid flowed down the horizontal channel and caused barrier failures in 5 of the 16 experiments. The photos of a typical magenta antifuse test device in operation in Figure 6.7B show that when enough excess fluid is added to cause a barrier failure, the failure occurs in an unpredictable location and likely would cause a catastrophic failure of the device.

In summary, these results show that **placing yellow antifuse barriers at locations where overflow is permissible can help protect other parts of the device where overflow must be avoided**.

6.3 Results

The results show that designers of paper microfluidic devices can use different ColorQube ink colors to impart specific behaviors to their devices. Among the single-color inks, it was observed that magenta had the fewest barrier failures, a finding that is consistent with previous work.[93] But it was also found that certain ink color combinations perform even better than magenta: the combinations CY, MK, YK, CMY, CYK and MYK had no barrier failures in this study. Therefore, for applications requiring the strongest possible hydrophobic barrier, researchers are advised to use these combinations.

This study was limited to a single model of wax-based color printer, the Xerox ColorQube 8570 printer shown in Figure 6.1. But based on the apparent similarity of the inks used in the different ColorQube printers, it is expected that these results will apply to other ColorQube models as well. Additionally, one can predict which colors will provide the strongest fluid barriers on a new printer without repeating all of these experiments. Based on the finding that ink mass correlates with barrier effectiveness (except for K), one can easily create same-size printed samples of all 15 ink color combinations, weigh them, and determine which non-K color combinations result in the largest amount of ink delivered to the paper (and are therefore most likely to provide the strongest barrier to fluid flow).

The one notable exception to the observation that increased ColorQube ink mass results in stronger fluid barriers was black, which had one of the largest on-paper ink masses but provided one of the weakest barriers to fluid flow. Why is the behavior of black ink so different from the other colors? The exact chemical compositions of the ColorQube inks are trade secrets, and reverse-engineering Xerox's products is beyond the scope of this work. However, if one were to predict which one of the ColorQube's four ink colors might behave very differently than the others, that color would undoubtedly be black. Unlike cyan, magenta, and yellow, which form a light and transparent coating on the paper to allow the reflectiveness of the underlying white paper to show through, black must form a dense and opaque layer that blocks all light from the paper. Consequently, black ink would be expected to contain a greater amount of pigments and dyes than other colors of ink, which may explain why the mass of black ink delivered to the paper is greater than the mass of almost all other colors (single and combinations). And if these black colorants are less hydrophobic than wax (or if they occupy space that would have otherwise been filled by hydrophobic wax), then it is reasonable to expect that black ColorQube ink will provide a less-hydrophobic and weaker barrier to fluid flow than the other color inks, which is exactly what was observed in this work.

Chapter 7

Rapid development and optimization of paper microfluidic designs

In this chapter, the software was used to automatically generate all of the device designs needed in the development of a real-world paper microfluidic assay (a test for detecting glucose and protein in urine). Building on an earlier tool that allowed for selecting among a few pre-configured paper microfluidic device designs, each having only one or two user-selectable parameters [86], the software enables the user to specify *all* device design parameters. The software then determines a valid device configuration, generates the device geometry, and outputs a custom graphics file ready for fabrication and use. Additionally, the framework uses the user's specifications of the device design plus the characteristics of the paper used to predict the total volume of fluid needed to operate the generated paper microfluidic device. Using this software, the designer of a complex paper microfluidic device (like the the fluidic multiplexer described above) could generate a new design (*e.g.*, with a different number of selectable channels) by simply changing a couple lines of code. Researchers developing their own paper microfluidic assays can use a web-based version of the software provided at ”`http://www.binary7design.com/rfa_trial/`”.

7.1 Materials and Methods

7.1.1 Supported paper microfluidic device fabrication techniques

The software framework presently targets the Xerox ColorQube line of wax-ink color printers, which others have previous used for paper microfluidic device fabrication. These printers contain solid blocks of wax-based ink which the printer melts and deposits on the paper. Chandler *et al.* [12] showed that since this wax ink is hydrophobic, it can be used to make channel walls and define fluid paths in paper microfluidics. The print process uses four primary colors of printers: cyan (C), magenta (M), yellow (Y), and black (K)[58] which can be used in various combinations to produce full color images. The software allows users to customize the ink colors in the generated paper microfluidic devices, for use as a color code, or for controlling the behavior of the chip using color-specific behavior. The files output by the software are intended for use by these printers, although they could be adapted for use in other paper microfluidic fabrication techniques like computer-controlled cutting tools.

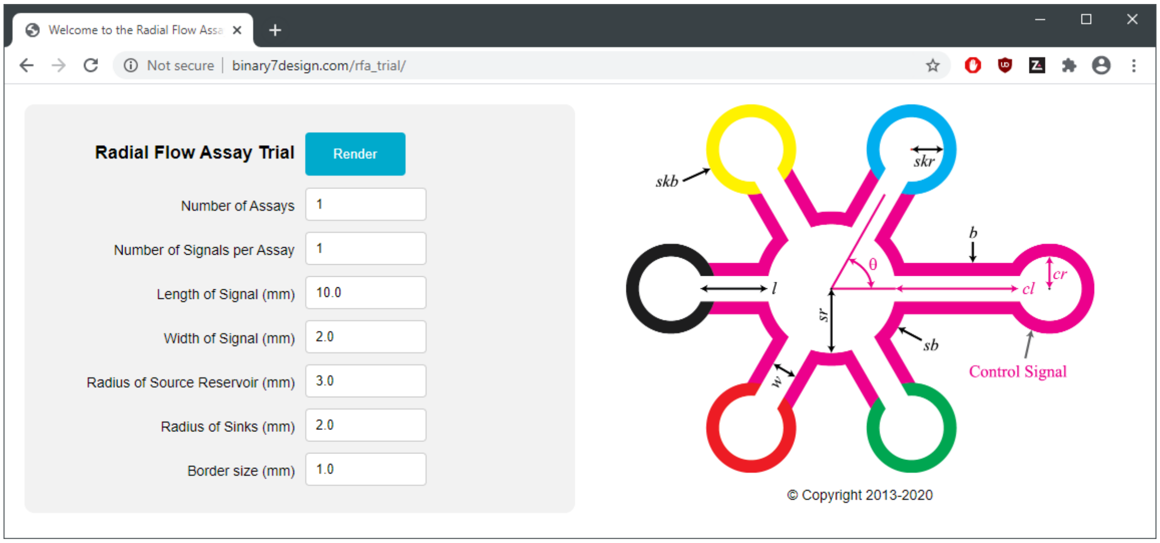
7.1.2 Using the software

The software framework features several pre-built devices that researchers can use to perform a variety of assays by specifying a small number of parameters for their desired application. Additionally, the framework provides a parameterizable component library including channels and reservoirs that a user can utilize to assemble a wide variety of multi-component devices for different fluid usage and evaluation scenarios. Once a device has been configured for usage, the user can select any of the standard file formats for output, such as PDF for printing, SVG for printing or image usage, or DXF to drive machines that plot lines or cut substrate materials.

To evaluate the software framework, the ***radial flow assay (RFA)*** component from the library was used for this work. The RFA (Fig. 7.1) consists of a central circular source reservoir where a sample is added to the device, one or more sink reservoirs where assay reagents are located, and a "control" sink that is further from the source than the other sinks. If fluid added to the source reservoir flows all the way to the end of the control sink, then it is likely that the fluid also traveled the (shorter) distances to all the other sink reservoirs. The RFA component features multiple configuration options from which the user may choose, including:

- **The number of assays desired** and **the number of copies of each assay to include**, which allow the researcher to specify multiple tests and runs per test, all from a single sample.

- The **source reservoir**, where fluid under test is deposited, has a **radius** sr, which determines its overall fluid volume capacity, and a **border width** sb, which ensures


Welcome to the Radial Flow Assa
Not secure | binary7design.com/rfa_trial/
Render
skb
skr
b
θ
cr
cl
l
sr
sb
w
Control Signal
© Copyright 2013-2020

Radial Flow Assay Trial

Number of Assays: 1

Number of Signals per Assay: 1

Length of Signal (mm): 10.0

Width of Signal (mm): 2.0

Radius of Source Reservoir (mm): 3.0

Radius of Sinks (mm): 2.0

Border size (mm): 1.0

Figure 7.1: **Online web interface.** The framework's Radial Flow Assay (RFA) component includes several user-customizable parameters, including the radius sr and border width sb of the source reservoir, the radius skr and border width skb of the sink reservoir, the length l and width w of the channels, and the overall border width b of the assay. Once these are specified by the user, the framework then calculates the angle θ of the channels, the minimum necessary channel length cl of the control channel, and the minimum necessary channel radius cr for the control reservoir.

proper containment of fluid delivered as the size of the source increases in volume.

- Each **sink reservoir** has the same parameters as the source reservoir regarding size and borders. The sinks are **color-coded** to provide an easy visual reference during runtime for the user to monitor the progress and results.

- The **channels** that connect the source to each sink are parameterized in terms of **length** l, **width** w, and **border width** b. These parameters influence the total volume of fluid required per assay, the per-assay runtime, and the total device runtime.

After a user specifies these options, the framework then calculates the angle θ between the channels, as well as the minimum necessary channel length cl and channel radius *cr* for the control channel and reservoir based on the length of the longest channel and largest sink value specified by the user.

The software framework defaults to a magenta (M) wax barrier, which was previously determined to be the most effective single ink option for fluid containment using printed wax barriers [84]. Each assay/sink pairing has a color assigned to it from a selection of predetermined colors to aid in identification of which assay is being performed in the device: cyan (C), yellow (Y), black (K), red (MY), green (CY), and blue (CM).

Once a device is configured, the software framework then generates a graphics file ready to print on the color printer. In this manner, over 50 different device designs used in this study were easily generated in less than a second per device.

7.1.3 Fabricating automatically designed paper microfluidic devices

All paper microfluidic device designs that were generated by the software were fabricated via a standard process that utilizes a wax-ink-based color printer. Two types of device testing were conducted in this study: flow testing (using dyes to visualize the paths followed by fluid in the devices) and chemical analysis (a simulated urinalysis assay). Finalized designs were output as PDF files that were then sent to a Xerox ColorQube 8570 wax-based color printer.

To calculate the fluid capacity of the paper, the framework needs to be informed regarding the thickness and porosity of the paper being used in the physical device. The paper used was “101” fast qualitative filter paper (“Lab Nerd” brand; Amazon.com) with a measured thickness of a single sheet (100 μm). A 50% porosity was assumed, meaning that 50% of the paper volume could be occupied by fluid. As this characteristic of the paper is particular to each brand and type, the number must be available to the framework for accurate results. Determining actual porosity can involve microscopic inspection when not supplied by the manufacturer, but the default of 50% serves as useful starting point. The paper was trimmed to 200 × 120 mm for manual feeding into the printer. Once printed, each layout was re-heated, printed side up, 5–7 cm above a 166 °C laboratory hotplate (temperature measured using an infrared thermometer) until wax melting was observed via color fading and blurring of printed edges. Reheating the wax re-melts the ink so that it permeates the full thickness of the paper, which ensures an effective hydrophobic barrier throughout. As a final step, a self-adhesive PCR plate sealing film (Microseal B; Bio-Rad, Hercules, CA) was added to the reverse side of the printed piece, adhering both to the paper and the wax barriers, thereby prohibiting fluid travel outside of the specified fluid boundaries.

7.1.4 Testing automatically designed paper microfluidic devices

For testing the flow of liquids through automatically designed paper microfluidics, deionized water containing a small amount of food coloring (3 drops per 50 mL) was delivered to the source reservoir of each device using a pipette. Each test run was photographed for the entire duration from initial delivery to completion and/or failure of the device due to either drying or wax barrier failure. For performing chemical analysis on automatically design paper microfluidics devices, a modified urinalysis protocol was used based on the one used by Martinez *et al.* [71] but with an additional analyte added. The test detects glucose, protein, and nitrite in a urine sample.

- Paper pads for detecting glucose were prepared by first depositing 2 μL of 0.6 M potassium iodide and waiting for a few minutes to allow the paper to dry. Next, 2 μL of a solution was added to the paper containing 700 units of horseradish peroxidase per mL (Thermo Scientific) and 2500 units of glucose oxidase per mL (MP Biomedicals) in 0.1 M potassium phosphate buffer at pH 6.0, then waited until the paper was dry before using the devices. The glucose tests are brown in the presence of glucose and colorless otherwise.

- Paper pads for detecting protein were prepared by first depositing 2 μL of 250 mM citrate buffer at pH 1.8 and waiting for the paper to dry. Then, 2 μL of a 3.3 mM solution was added of tetrabromophenol blue in 95% ethanol, then waited until the paper was dry before using the devices. The protein tests are blue in the presence of protein and yellow otherwise.

- Paper pads for detecting nitrite were prepared by depositing 2 μL of Griess reagent (Sigma-Aldrich), then waiting until the paper was dry before using the devices. The nitrite tests are pink in the presence of nitrite and colorless otherwise.

To test these devices, eight simulated urine samples were prepared: one with glucose (167 mM) in water; one with bovine serum albumin (BSA; 10 μM) in water; one with sodium nitrite (500 mM) in water; three samples containing all pairwise combinations of the three analytes; one sample containing all three analytes; and one "control" sample containing only water with no added glucose or BSA. For each test, 36 μL of simulated urine were added to the Input reservoirs of each paper microfluidic device. The remainder of the fluid in the simulated urine was initially 1x phosphate-buffered saline (PBS) in Phase

9 of this study, but was changed to water in Phase 10 after an undesired side reaction was observed between PBS and two of the urinalysis reagents (details in *Results and Discussion* Phase 9 below).

7.2 Results and discussion

The software framework was used to generate all of the paper microfluidic device designs needed to develop a model urinalysis assay, thereby validating its practical efficacy. This required 51 different device designs which were used in a total of 120 lab experiments. This section provides a chronological account of the interactions with the software framework during this development process.

7.2.1 Phase 1: Finding optimal wax ink border width

The first step in designing the model urinalysis assay was to determine an appropriate width for the wax ink border (b in Fig. 7.1) that contains the fluid flow paths. A border that is too narrow may not effectively block fluid flow and cause loss of containment of the fluid, and a border that is too wide unnecessarily wastes both wax ink and space on the device.

In this phase, the border width parameter in the framework was used to generate device designs with border widths of 0.5, 1.0, 1.5, and 2.0 mm. The devices were otherwise identical, each having a central Source reservoir with magenta-colored borders where fluid is added to the device, a single Sink reservoir with cyan-colored borders where detection of an analyte would occur, and a single Control reservoir with magenta-colored borders which

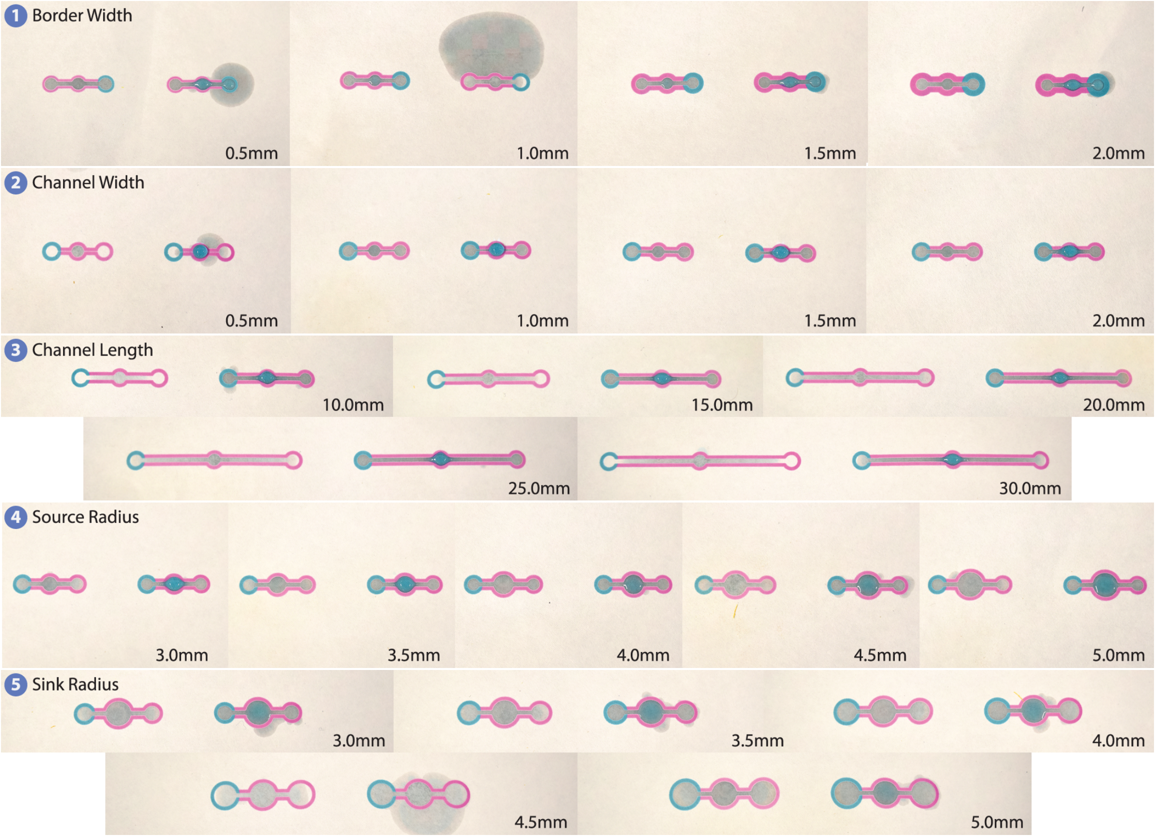


Figure 7.2: **Phases 1-5** Results from testing paper microfluidic device designs generated by the software framework, for optimizing ink border width (Phase 1), channel width (Phase 2), channel length (Phase 3), source radius (Phase 4), and sink radius (Phase 5).

will later be used to confirm that adequate fluid volume has been added to the device during use. The framework predicted the same fluid capacity for all four device designs, 8.4 μL, because changing the border width does not affect the volume of fluid contained inside the border.

As shown in Fig. 7.2 (Phase 1), the software generated two copies of each device design for evaluation. During testing, the left-hand copy of each device received 9.0 μL of fluid, and the right-hand copy received 50.0 μL to observe effects from a user inadvertently adding too much fluid to the device.

It was found that all four border widths successfully contained the smaller 9.0 μL volumes. However, all four barrier widths *failed* to contain the larger 50.0 μL volumes, with the largest fluid loss associated with the thinnest barriers (0.5 and 1.0 mm) and reduced fluid loss associated with the thicker barriers (1.5 and 2.0 mm). Additionally, failures typically occurred at cyan-colored borders; this indicates that cyan ink provides a weaker barrier to flow than magenta ink, a observation consistent with earlier studies [84].

Based on these results, 1.0 mm was chosen as the wax barrier width to use in subsequent devices. This determination required eight experiments using four different device designs generated by the software framework. The experimental parameters used in Phase 1 are summarized in Table 7.1.

7.2.2 Phase 2: Finding optimal fluid channel width

After finalizing the wax ink barrier width for the model urinalysis device in Phase 1, the channel width parameter was varied to generate a series of test device designs to

Table 7.1: **Experimental parameters used during Phase 1 (optimization of wax ink fluid barrier width; Fig. 7.2).**

Border Width (mm)	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
0.5	8.4	9.0 (Pass)	50.0 (Over)
1.0	8.4	9.0 (Pass)	50.0 (Over)
1.5	8.4	9.0 (Pass)	50.0 (Over)
2.0	8.4	9.0 (Pass)	50.0 (Over)

Border widths were varied from 0.5 to 2.0 mm. All devices had one Source reservoir with radius 2.5 mm, one Sink reservoir with radius 2.5 mm, one Control reservoir with radius 2.5 mm, a 5.0 mm channel length, and 2.0 mm channel width. Results were categorized as having executed successfully (Pass), or having failed either due to under-filling (Under) or overflowing fluid barriers (Over).

investigate the optimal fluid channel width (w in Fig. 7.1). Channel width is an important parameter in part because the post-printing heating of a paper microfluidic device (described above in *Materials and Methods*) causes the wax ink features to spread slightly as the wax melts. A device that is printed with an excessively narrow channel can have these channels sealed closed when the wax melts. Conversely, channels that are too wide waste space on the device and consume additional fluids and reagents.

The software framework was used to generate device designs with channel widths of 0.5, 1.0, 1.5, and 2.0 mm. All devices had the same overall design as in Phase 1 (one Source reservoir for fluid, one Sink reservoir for detection, and one Control reservoir for verifying adequate loading of the device). Additionally, all devices had the 1.0 mm wide wax ink borders and was found to be optimal in Phase 1. The software predicted fluid capacities for the channel widths of: 6.5 μL @ 0.5 mm, 7.1 μL @ 1.0 mm, 7.8 μL @ 1.5 mm, and 8.4 μL @ 2.0 mm.

The software was again used to generated two copies of each device design, as shown in Fig. 7.2 (Phase 2). During testing, the left-hand copy of each device received 9 μL of fluid, and the right-hand copy received 50 μL of fluid (again, chosen to simulate a user accidentally adding excess fluid to the device).

In the devices with the channel width of 0.5 mm, fluid added to the Source reservoir failed to completely fill the Sink and Control reservoirs. When a small amount of fluid was added to these devices (9 μL), it remained in the Source reservoir, demonstrating that the 0.5 mm wide channels are too narrow for reliable fluid flow. When an excess amount of fluid was added (50 μL), some fluid did reach the Sink and Control reservoirs, but fluid also overflowed the wax barriers. These results indicate that 0.5 mm wide channels are too narrow for reliable use in the urinalysis device. However, all devices with wider channels (1.0, 1.5, and 2.0 mm) worked as intended, filling the Sink and Control reservoirs without losing containment of the fluid.

Based on these results, 2.0 mm was chosen as the channel width to use in subsequent experiments. This phase of development required eight experiments using four different device designs generated by the software. These experimental parameters are summarized in Table 7.2.

Table 7.2: **Experimental parameters used during Phase 2 (optimization of fluid channel width; Fig. 7.2).**

Channel Width (mm)	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
0.5	6.5	9.0 (Under)	50.0 (Over)
1.0	7.1	9.0 (Pass)	50.0 (Pass)
1.5	7.8	9.0 (Pass)	50.0 (Pass)
2.0	8.4	9.0 (Pass)	50.0 (Pass)

Border widths were varied from 0.5 to 2.0 mm. All devices had one Source reservoir with radius 2.5 mm, one Sink reservoir with radius 2.5 mm, one Control reservoir with radius 2.5 mm, a 5.0 mm channel length, and 2.0 mm channel width. Results were categorized as having executed successfully (Pass), or having failed either due to under-filling (Under) or overflowing fluid barriers (Over).

7.2.3 Phase 3: Finding optimal channel length

With optimal values for border width and channel width found, the software framework was used to explore the effects of channel length (l in Fig. 7.1) on device operation. Channels are necessary to provide fluid transport; however, if they are too long, the fluid may be unable to travel to the end of the channel due to backflow pressure or the leading edge of the fluid drying and forming a barrier [42]. Either case may result in the fluid being unable to completely fill the device to the desired volume. Conversely, channels that are too short may experience barrier failure due to overflow [49]. The software generated designs by specifying channel lengths of 10, 15, 20, 25, and 30 mm in the framework. Aside from the different channel lengths, these devices were identical to those used in Phase 2, with 1.0 mm wide wax ink borders and 2.0 mm wide channels. The framework calculated volumes for each channel length of 10.4 μL @ 10.0 mm, 12.4 μL @ 15.0 mm, 14.4 μL @ 20.0 mm, 16.4 μL @ 25.0 mm, and 18.4 μL @ 30.0 mm.

Again, the software generated the two copies of each device (Fig. 7.2 Phase 3). The left-hand copies received fluid volumes that were roughly equal to the predicted capacities of the devices: 11 μL @ 10.0 mm, 13.0 μL @ 15.0 mm, 15 μL @ 20.0 mm, 17.0 μL @ 25.0 mm, and 19.0 μL @ 30.0 mm. The right-hand copies all received 50 μL of fluid (a volume larger than the predicted capacities of all of the devices).

The results shown in Fig. 7.2 (Phase 3) provide several insights into both the device design and the software framework. For the left-hand devices that received fluid volumes roughly equal to the predicted capacity of the devices, none of the devices filled completely (that is, the fluid delivered to the Source reservoir did not quite reach the Sink and Control reservoirs). This suggests that the calculated volume of fluid alone does not necessarily guarantee successful completion. As the fluid travels further from the source, additional dynamics such as the aforementioned drying and backpressure begin to dominate capillary action [63] and would need to be taken into account by the researcher by increasing the volume of the liquid delivered (and ultimately addressed within the software framework to account for the additional fluid dynamics). Future versions of the software framework will allow for user-specified equations to be used for calculation and simulation of fluid volumes and behaviors. In contrast, for the right-hand devices that received excessively-large fluid volumes, all of the devices filled completely. Additionally, only one of the right-hand devices failed to contain the large volume delivered: fluid overflowed the borders of the smallest device (channel length = 10 mm). All other devices contained the large (50 μL) volumes without overflowing.

Based on these results, it was decided to continue using the 5.0 mm channel length to mitigate device failures due to the fluid dynamics that begin to assert themselves at longer lengths. This phase required ten experiments using five different device designs generated by the software. These experimental parameters are summarized in Table 7.3.

Table 7.3: **Experimental parameters used during Phase 3 (optimization of fluid channel length; Fig. 7.2).**

Channel Length (mm)	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
10.0	10.4	11.0 (Under)	50.0 (Over)
15.0	12.4	13.0 (Under)	50.0 (Pass)
20.0	14.4	15.0 (Under)	50.0 (Pass)
25.0	16.4	17.0 (Under)	50.0 (Pass)
30.0	18.4	19.0 (Under)	50.0 (Under)

Channel lengths were varied from 10.0 to 30.0 mm. All devices had one Source reservoir with radius 2.5 mm, one Sink reservoir with radius 2.5 mm, one Control reservoir with radius 2.5 mm, a 1.0 mm border width, and a 2.0 mm channel width.

7.2.4 Phase 4: Finding optimal Source reservoir radius

In this phase of device design, the software framework was used to aid in finding an optimal radius for the devices' Source reservoirs (*sr* in Fig. 7.1). A larger Source reservoir provides an easier target for a user delivering fluid to the paper microfluidic device. However, as the Source grows larger, it retains more fluid that does not flow to the Sink reservoirs for testing; this effectively wastes sample. So, the ideal Source radius is a balance between user-friendliness and sample conservation.

The software framework was used to generate device designs varying the Source reservoir radii to the values of 3.0, 3.5, 4.0, 4.5, and 5.0 mm. The other device parameters were unchanged from the optimal values found in the previous phases: 1.0 mm wide wax ink borders, 2.0 mm wide fluid channels, and 5.0 mm long fluid channels. The software-predicted fluid capacities of these devices of 9.3 μL @ 3.0 mm, 10.3 μL @ 3.5 mm, 11.4 μL @ 4.0 mm, 12.8 μL @ 4.5 mm, and 14.3 μL @ 5.0 mm.

As in Phase 3, the software generated two copies of each device design shown in Fig. 7.2 (Phase 4), with the left-hand copy receiving a volume roughly equal to the predicted capacity of the device: 10.0 μL @ 3.0 mm, 11.0 μL @ 3.5 mm, 12.0 μL @ 4.0 mm, 13.0 μL @ 4.5 mm, and 15.0 μL @ 5.0 mm. The right-hand copy received a volume of 50.0 μL, to greatly exceed the volumes of the devices.

The results in Fig. 7.2 (Phase 4) show that all devices filled to completion, regardless of Source reservoir size. Some minor leakage through the wax barriers was observed, particularly in the right-hand devices that received excessively-large volumes of fluid. However, the amount of leakage was very small and all five Source reservoir radii were judged as successful. Additionally, as expected, the larger Sources were the best targets for fluid delivery. Based on these results, in subsequent tests the Source reservoir radii of 2.5 mm (for experiments utilizing smaller volumes of fluid) or 5.0 mm (for larger volumes of fluid) was ysed. This phase required ten experiments using five different device designs generated by the software. These experimental parameters are summarized in Table 7.4.

Table 7.4: **Experimental parameters used during Phase 4 (optimization of Source reservoir radius; Fig. 7.2).**

Source Radius (mm)	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
3.0	9.3	10.0 (Pass)	50.0 (Pass)
3.5	10.3	11.0 (Pass)	50.0 (Pass)
4.0	11.4	12.0 (Pass)	50.0 (Pass)
4.5	12.8	13.0 (Pass)	50.0 (Over)
5.0	14.3	15.0 (Over)	50.0 (Pass)

Source reservoir radii were varied from 3.0 to 5.0 mm. All devices had one one Sink reservoir with radius 2.5 mm, one Control reservoir with radius 2.5 mm, a 1.0 mm border width, a 2.0 mm channel width, and a 5.0 mm channel length. While some leakage across the wax ink barriers were observed, this leakage was judged to be minor and categorized all Source radii as passing the test.

7.2.5 Phase 5: Finding optimal Sink reservoir radius

In Phase 5, the software was used to also explore optimal values for the Sink reservoir radius (*skr* in Fig. 7.1). This test was intended to both confirm that the observed behavior of Source reservoirs is also consistent with Sink reservoir behavior, and observe the Sink reservoir in isolation from the fluid delivery to see if there might be failures in the destination of fluid that might adversely affect any chemical reactions occurring in the Sink reservoirs.

By specifying the radii of the Sinks reservoirs used, device designs with radii of 3.0, 3.5, 4.0, 4.5, and 5.0 mm were generated. The Source reservoir radii were held constant at 5.0 mm, and the other device features were unchanged from Phase 4. The software predicted device fluid capacities of: 16.1 μL @ 3.0 mm, 18.3 μL @ 3.5 mm, 20.7 μL @ 4.0 mm, 23.5 μL @ 4.5 mm, and 26.6 μL @ 5.0 mm.

As before, two copies of each device were generated: the left-hand copy received approximately the volume predicted by the software of: 17.0 μL @ 3.0 mm, 19.0 μL @ 3.5 mm, 21.0 μL @ 4.0 mm, 24.0 μL @ 4.5 mm, and 27.0 μL @ 5.0 mm. The right-hand copy received 50 μL which is much larger than the capacity of each device.

The results shown in Fig. 7.2 (Phase 5) show that devices with all five Sink sizes filled completely, for all volumes of fluid added to the devices. However, all of the devices that received excess (50 μL) volumes of fluid demonstrated leakage over the wax ink fluid barrier. Based on these results, it was decided to maintain the 2.5 mm Sink radius for subsequent experiments. The size of Sink reservoirs is ultimately dependent on the amount of fluid and reagents needed for successful operation of the device, and this trial demonstrates the ease with which the size of the Sink reservoir can be adjusted using the software to accommodate users' needs. The experimental parameters are summarized in Table 7.5.

Table 7.5: **Experimental parameters used during Phase 5 (optimization of Sink reservoir radius; Fig. 7.2).**

Sink Radius (mm)	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
3.0	16.1	17.0 (Pass)	50.0 (Over)
3.5	18.3	19.0 (Pass)	50.0 (Over)
4.0	20.7	21.0 (Pass)	50.0 (Over)
4.5	23.5	24.0 (Under)	50.0 (Over)
5.0	26.6	27.0 (Pass)	50.0 (Over)

Sink reservoir radii were varied from 3.0 to 5.0 mm. All devices had one one Source reservoir with radius 5.0 mm, one Control reservoir with the same radius as the Sink reservoir, a 1.0 mm border width, a 2.0 mm channel width, and a 5.0 mm channel length. Although overflow failure was observed in Source reservoirs, all sizes of Sink reservoirs were capable of receiving the increase of fluid. Barrier weaknesses like these in a device design can be easily addressed by increasing barrier widths in the software framework.

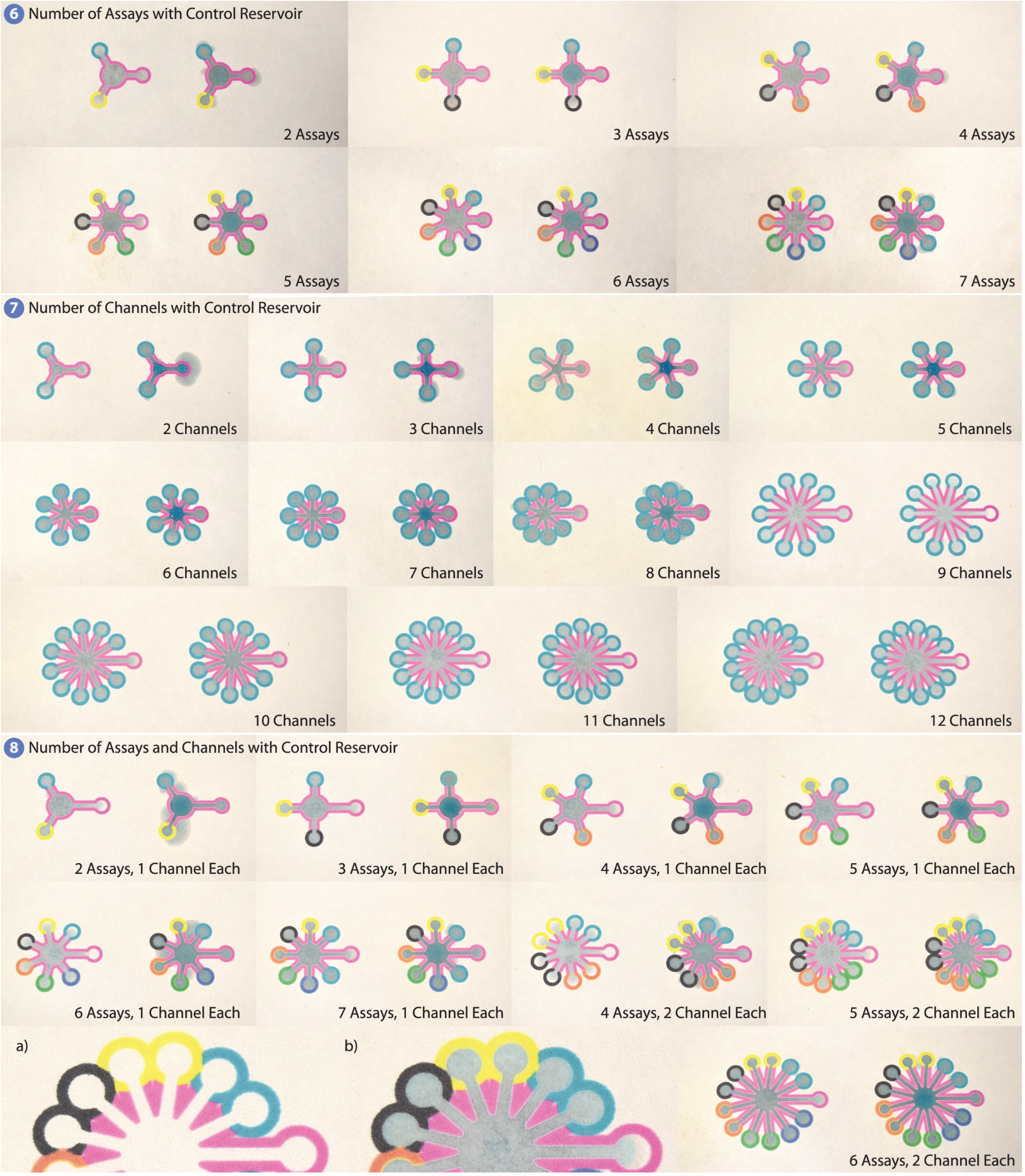


Figure 7.3: **Phases 6-8** Further results from testing paper microfluidic device designs generated by the software framework, for optimizing the number of assays (Phase 6), the number of channels (Phase 7), and the combination of assays and channels (Phase 8). The closeups in 8a and 8b show a device with insufficient separation between sinks; this is easily remedied in the software by increasing the channel length.

7.2.6 Phase 6: Finding optimal number of assays

At this point in the study, the software framework had generated all of the device designs necessary to find optimal values for the wax ink border width (b in Fig. 7.1), the fluid channel width (w), the fluid channel length (l), the source reservoir radius (sr), and the sink reservoir radius (skr) – without manual intervention of the user to redraw and refactor the design under test. These parameters define the low-level design of the desired paper microfluidic device. The next step in development explores the high-level design of the device—the number of assays it performs, and the number of channels per assay.

In Phase 6, designs intended to perform multiple assays were tested on a single sample. Each assay would be carried out in its own Sink reservoir. Performing multiple assays in parallel is generally advantageous because it provides the user with more information about the same sample, but adding additional Sink reservoirs to the device increases the fluid volume requirements and overall complexity of the device. An optimal number of assays would be a balance between these two constraints.

Using the software framework, 2, 3, 4, 5, 6, and 7 parallel assay designs were specified which were then output for evaluation (Fig. 7.3, Phase 6). Each design also incorporated a single Control reservoir for confirming successful loading of fluid into the device. The software framework automatically assigns a different wax ink color to the borders surrounding each assay's Sink reservoir (a new feature for this work); aiding the user in interpreting the results of a test. The other device parameters were those determined as optimal during Phases 1 through 5 (b = 1.0 mm, w = 2.0 mm, l = 5.0 mm, sr = 5.0 mm, and skr = 2.5 mm). The software predicted device fluid capacities of: 17.2 μL for 2 assays, 20.2 μL for 3 assays, 23.2 μL for 4 assays, 26.1 μL for 5 assays, 29.1 μL for 6 assays, and 32.1 μL for 7 assays.

As before, the software generated two copies of each device for fabrication. The left-hand devices in Fig. 7.3 (Phase 6) received a volume of fluid roughly equal to that predicted by the software for each design: 18.0 μL for 2 assays, 21.0 μL for 3 assays, 24.0 μL for 4 assays, 27.0 μL for 5 assays, 30.0 μL for 6 assays, and 33.0 μL for 7 assays. The right-hand devices again received an excessive volume of 50 μL.

The results from testing these devices (Fig. 7.3 Phase 6) show that all designs filled with fluid successfully. Some minor leakage across the wax ink barrier was observed in the devices that received excess (50 μL) fluid, but in general the devices performed well regardless of the number Sink reservoirs for assays on the device. Based on these results, it was determined that the device can support at least 7 parallel assays without negatively impacting the performance of the device. These experimental parameters are summarized in Table 7.6.

7.2.7 Phase 7: Finding optimal number of channels

In this phase, the optimal number of channels for each assay in the paper microfluidic device was explored. Adding channels can provide redundancy (in case some assays fail) and statistical confidence. However, channels also occupy space in the device, leaving less space for additional assays. The optimal number of channels is a balance between these two requirements. The software framework was used to generate device designs shown in Fig. 7.3 (Phase 7) containing 2 to 12 copies of a single sink reservoir (plus a Control reservoir).

Table 7.6: **Experimental parameters used during Phase 6 (optimization of number of parallel assays; Fig. 7.3).**

# of Assays	Channels per Assay	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
2	1	17.2	18.0 (Pass)	50.0 (Over)
3	1	20.2	21.0 (Pass)	50.0 (Pass)
4	1	23.2	24.0 (Pass)	50.0 (Pass)
5	1	26.1	27.0 (Under)	50.0 (Pass)
6	1	29.1	30.0 (Pass)	50.0 (Pass)
7	1	32.1	33.0 (Pass)	50.0 (Over)

All devices had one one Source reservoir with radius 5.0 mm, Sink reservoirs with radius 2.5 mm, one Control reservoir with the same radius as the Sink reservoir, a 1.0 mm border width, a 2.0 mm channel width, and a 5.0 mm channel length.

The devices with 2 to 7 sinks had constant channel lengths of 5.0 mm. For devices with 8 or more sinks, the software automatically increases the length of the channel leading to the Control sink to place it outside of the radius of the other sinks. This provided more room for additional channel sinks, and also helps the Control sink function as an actual control: if fluid successfully completes the (longer) path to the Control sink, then the user can be confident that the fluid also completed the (shorter) paths to the assay sinks. Additionally, the devices having 9 to 12 sinks had longer channels lengths of 10 mm for the assay sinks and 15 mm channel length for the Control sink. The length of the Control sink channel was calculated by the framework using the length of the longest channel in the device and adding the diameter of the largest Sink reservoir (in this case, 10 mm + 2 × 2.5 mm = 15 mm). These longer channel lengths provide additional room for more assay channels. Apart from these design details, the other device parameters were unchanged from Phase 6: $b =$ 1.0 mm, $w =$ 2.0 mm, $sr =$ 2.5 mm, and $skr =$ 2.5 mm. The software predicted device fluid capacities of: 11.4 μL for 2 channels, 14.3 μL for 3 channels, 17.3 μL for 4 channels, 20.2 μL for 5 channels, 23.2 μL for 6 channels, 26.2 μL for 7 channels, 30.0 μL for 8 channels, 43.0 μL for 9 channels, 47.0 μL for 10 channels, 50.9 μL for 11 channels, and 54.9 μL for 12 channels.

As before, two copies of each device were generated, with the left-hand copy receiving a fluid volume comparable to the predicted capacity of the device of: 12.0 μL for 2 channels, 15.0 μL for 3 channels, 18.0 μL for 4 channels, 21.0 μL for 5 channels, 24.0 μL for 6 channels, 27.0 μL for 7 channels, 31.0 μL for 8 channels, 43.0 μL for 9 channels, 48.0 μL for 10 channels, 51.0 μL for 11 channels, and 54.0 μL for 12 channels. The right-hand copy again receiving 50 μL.

The results in Fig. 7.3 (Phase 7) demonstrated barrier overflow errors in the 2- and 3-channel devices when excess fluid was delivered. Additionally, a small barrier failure was observed in the 3-channel device and attributed to a fabrication error where the backing tape did not fully seal to the wax barrier. However, as the number of channels increased, overflow errors subsided and underflow errors increased. Also, for the devices with the largest number of channels (and the longer channels), even though the Sink reservoirs did successfully fill, the Control reservoirs did not fill; this indicates to the user that the devices did not successfully complete their operation. These results are consistent with those previously observed in Phase 3 (overflow in smaller devices and under-filling in larger devices) and they confirm the need to account for additional fluid dynamics in test development and in the software framework. These parameters are summarized in Table 7.7.

7.2.8 Phase 8: Finding an optimal combination of assays and channels

In the final phase before testing the devices in a chemistry application, the framework was used to generate devices containing different combinations of assay and channel numbers. Successful designs from this phase would be chosen for use as the simulated urinalysis assay in the following phases.

The software generated designs (shown in Fig. 7.3 Phase 8) supporting 2, 3, 4, 5, 6, and 7 assays with a single channel each, and 4, 5, and 6 assays with two channels each. Again, the automatic color coding of the assay sinks assists the user in identifying which sink corresponds to which assay and which channel. The other device parameters were unchanged from Phase 6: b = 1.0 mm, w = 2.0 mm, sr = 5.0 mm, and skr = 2.5 mm) excepting the addition of the control signal as in Phase 7. As before, two devices were prepared for each device design, with the left-hand copy receiving the fluid volume predicted by the software framework of: 18.1 μL for 2 assays and 1 channel, 21.1 μL for 3 assays and 1 channel, 24.1 μL for 4 assays and 1 channel, 27.0 μL for 5 assays and 1 channel, 30.0 μL for 6 assays and 1 channel, 33.0 μL for 7 assays and 1 channel, 35.9 μL for 4 assays and 2 channels, 41.9 μL for 5 assays and 2 channels, 47.8 μL for 6 assays and 2 channels. Another device was fabricated where the channel lengths were increased to 10 mm for the 6 assay and 2 channel device to avoid overlap and the new calculated volume was 60.8 μL. The right-hand copy receiving an excessively large 50 μL (with one exception, described below).

The results in Fig. 7.3 (Phase 8) show that device behavior is mostly consistent with behaviors seen in Phase 6. However, the addition of the control sink revealed that although the 50 μL delivered to each device tended to fill the sinks, that amount was no longer enough to also fill the control sink. This indicates that the fluid needed by the device was more than the volume predicted by the framework based solely on geometric considerations. As such, the aforementioned fluid dynamics of drying time, backpressure, etc., also need to be factored in to assure proper execution of devices. Additionally, when a 6 assay, 2 channels per assay device was tested, it was noticed that the device's 5.0 mm channel lengths caused the sinks to be placed so close together that they overlapped, as shown in Fig. 7.3 (Phase 8a). Although the device did run to completion in Fig. 7.3 (Phase 8b), it was nonetheless concluded that the lack of space between the Sink reservoirs could give this design a greater likelihood of cross-contamination between Sink reservoirs and make barrier failures more difficult to detect. Increasing channel length by doubling from 5.0 mm to 10.0 mm in the software framework eliminated this risk which then increased the fluid capacity of the device by 13 μL to 61 μL. Consequently, the amount of fluid delivered to each device was then increased to 63 μL for the left device to accommodate the increase proportionally to the previously delivered volumes, while the amount delivered to the right device was also doubled to ensure that a overflow failure would be possible. Otherwise, the fluid volume delivered to the devices with 5 mm channel lengths was: 19.0 μL for 2 assays and 1 channel, 22.0 μL for 3 assays and 1 channel, 25.0 μL for 4 assays and 1 channel, 28.0

μL for 5 assays and 1 channel, 30.0 μL for 6 assays and 1 channel, 33.0 μL for 7 assays and 1 channel, 36.0 μL for 4 assays and 2 channels, 42.0 μL for 5 assays and 2 channels, 48.0 μL for 6 assays and 2 channels. Again, 50 μL was delivered to the right device except for the aforementioned increase to 100.0 μL as noted.

7.2.9 Phase 9: Model urinalysis assay with three analytes

At this point, the software framework had generated 51 different designs that were used in 102 separate experiments to arrive at device design parameters that are optimal for this application. In the final phases of assay development, these newly discovered design parameters were used to generate paper microfluidic devices intended to detect clinically-relevant biomarkers in urine. In Phase 9 of assay development, the software was used to specify and produce a device design that supported three different assays (glucose, nitrite, and protein) with three replicates per assay, for a total of nine color-coded Sink reservoirs (glucose in cyan-bordered sinks, nitrite in yellow-bordered sinks, and protein in black-bordered sinks) plus a tenth Control reservoir (magenta-bordered) to confirm complete loading of fluid into the device. The device was tested using samples of synthetic urine consisting of 1x phosphate-buffered saline and containing the amounts of glucose, nitrite, and bovine serum albumin (BSA; a protein) described in the *Materials and Methods* section. Each device received 36 to 40 μL of synthetic urine and was left to sit for between 12 and 18 minutes before photographing the resulting color changes.

The results shown in Fig. 7.4 (Phase 9) show that while the software-optimized device design functioned as intended, the assay chemistry exhibited some unexpected problems. All nine assay sinks plus the Control sink filled successfully, and the glucose assay worked as expected, with a brown color appearing in the glucose-sensing sinks when glucose is present in the synthetic urine (device on left in Fig. 7.4 (Phase 9)) but no color change in the glucose-sensing sinks when glucose was absent (right in Fig. 7.4 (Phase 9)). However, while the expected blue color change was visible in protein-sensing sinks when BSA was present in the synthetic urine (left), a slight blue color was also evident in the protein-sensing sinks when BSA-free synthetic urine was added (right). Even though the magnitude of the color change was much greater for the BSA-containing sample than the BSA-free sample, this nonetheless could lead to "false positive" errors and had to be addressed. After additional experimentation, it was determined that some component of the PBS buffer (possibly the phosphate anion) seemed to be causing the unwanted color change in the protein-free samples, and in Phase 10 of the assay development the PBS from the synthetic urine was eliminated. Additionally, it was observed that a red color was visible in the nitrite-sensing sinks regardless of whether the synthetic urine sample contained nitrite anions. This was attributed to a poor choice of color-changing reagent: the Griess reagent used is typically used to detect nitrites in solution, not in a dried spot on paper, so in Phase 10 of assay development the nitrite measurement was abandoned.

7.2.10 Phase 10: Model urinalysis assay with two analytes

In the previous phase, the initial experiments performing a urinalysis assay with three analytes on paper microfluidic devices generated by the software illuminated some issues that needed addressing. Fixing one of these issues (the problematic Griess test for nitrite) required eliminating the test from the device, which would normally require redesigning the paper device. Leveraging the software, generating updated designs without the nitrite test was simple and fast.

Fig. 7.4 (Phase 10) shows the updated device design. Eliminating the nitrite test sinks provided enough room for four channels each of the remaining two tests (glucose and protein), organized into two sets of two. Again, the software uses the wax ink border color to provide a “key” for the assays: glucose testing in sinks with cyan and black borders, and protein testing in sinks with yellow and red borders. Additionally, in testing these devices a water-based synthetic urine was used (no phosphate anions).

The results from testing this device show that both tests functioned as intended: all four glucose-detecting sinks demonstrated a dark brown color change when glucose was present in the synthetic urine; all four protein-detecting sinks demonstrated a blue color change when BSA was present in the synthetic urine; and none of the test sinks changed color when their corresponding target was absent from the synthetic urine. Thus, the proof-of-concept device functioned as desired. Creating this device required 51 different device design iterations, all of which were generated automatically by the software framework.

In this chapter, the process of developing a paper microfluidic diagnostic was documented from an idea to an optimized and chemically-functional prototype, without manually designing a single device. Using the software framework, it was possible to automatically generate 51 device variations just through varying the newly introduced parametric capabilities. Fabricating and testing these automatically-generated designs took a total of 16 hours in the lab. In other words, with the assistance of the software framework, it was possible to go from an idea to a functional and optimized prototype in two business days.

Table 7.7: **Experimental parameters used during Phase 7 (varying the number of channels per assay; Fig. 7.3).**

# of Channels	Channel Length (mm)	Volume Calc'd. (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
2	5.0	11.4	12.0 (Pass)	50.0 (Over)
3	5.0	14.3	15.0 (Over)	50.0 (Over)
4	5.0	17.3	18.0 (Pass)	50.0 (Pass)
5	5.0	20.2	21.0 (Pass)	50.0 (Pass)
6	5.0	23.2	24.0 (Pass)	50.0 (Pass)
7	5.0	26.2	27.0 (Pass)	50.0 (Pass)
8	5.0	30.0	31.0 (Pass)	50.0 (Over)
9	**10.0**	43.0	43.0 (Pass)	50.0 (Under)
10	**10.0**	47.0	48.0 (Under)	50.0 (Pass)
11	**10.0**	50.9	51.0 (Under)	50.0 (Under)
12	**10.0**	54.9	54.0 (Under)	50.0 (Under)

All devices had one Source reservoir with radius 2.5 mm, Sink reservoirs with radius 2.5 mm, one Control reservoir with the same radius as the Sink reservoir, a 1.0 mm border width, a 2.0 mm channel width, and channel lengths of 5.0 mm for the devices with 2 to 8 channels and 1.0 mm for the devices with 9 to 12 channels. Overflow failure in 3 cases was observed in the lower volume tests, but observed under-fill errors as the number of channels increased. Note that although the Sink reservoirs did fill completely, the Control reservoir did not fill completely, which would indicate to the user that insufficient fluid volume has been added to the device.

Table 7.8: **Experimental parameters used during Phase 8 (varying both assays and channels; Fig. 7.3).**

# of Assays	# of Channels	Volume Calculated (μL)	Volume Delivered Left (μL)	Volume Delivered Right (μL)
2	1	18.1	19.0 (Under)	50.0 (Over)
3	1	21.1	22.0 (Pass)	50.0 (Pass)
4	1	24.1	25.0 (Under)	50.0 (Pass)
5	1	27.0	28.0 (Pass)	50.0 (Pass)
6	1	30.0	30.0 (Under)	50.0 (Over)
7	1	33.0	33.0 (Under)	50.0 (Pass)
4	2	35.9	36.0 (Under)	50.0 (Over)
5	2	41.9	42.0 (Under)	50.0 (Over)
6	2	47.8	48.0 (Under)	50.0 (Under)
6	2	60.8	63.0 (Pass)	**100.0** (Pass)

Assay count was varied from 2 to 6 and channel count was either 1 or 2. All devices had one Source reservoir with radius 5.0 mm, Sink reservoirs with radius 2.5 mm, one Control reservoir with the same radius as the Sink reservoir, a 1.0 mm border width, a 2.0 mm channel width, and a default channel length of 5.0 mm initially until the number of channels increased to 12, after which the channel length was increased to 10 mm to accommodate the additional Sink reservoirs without overlap.

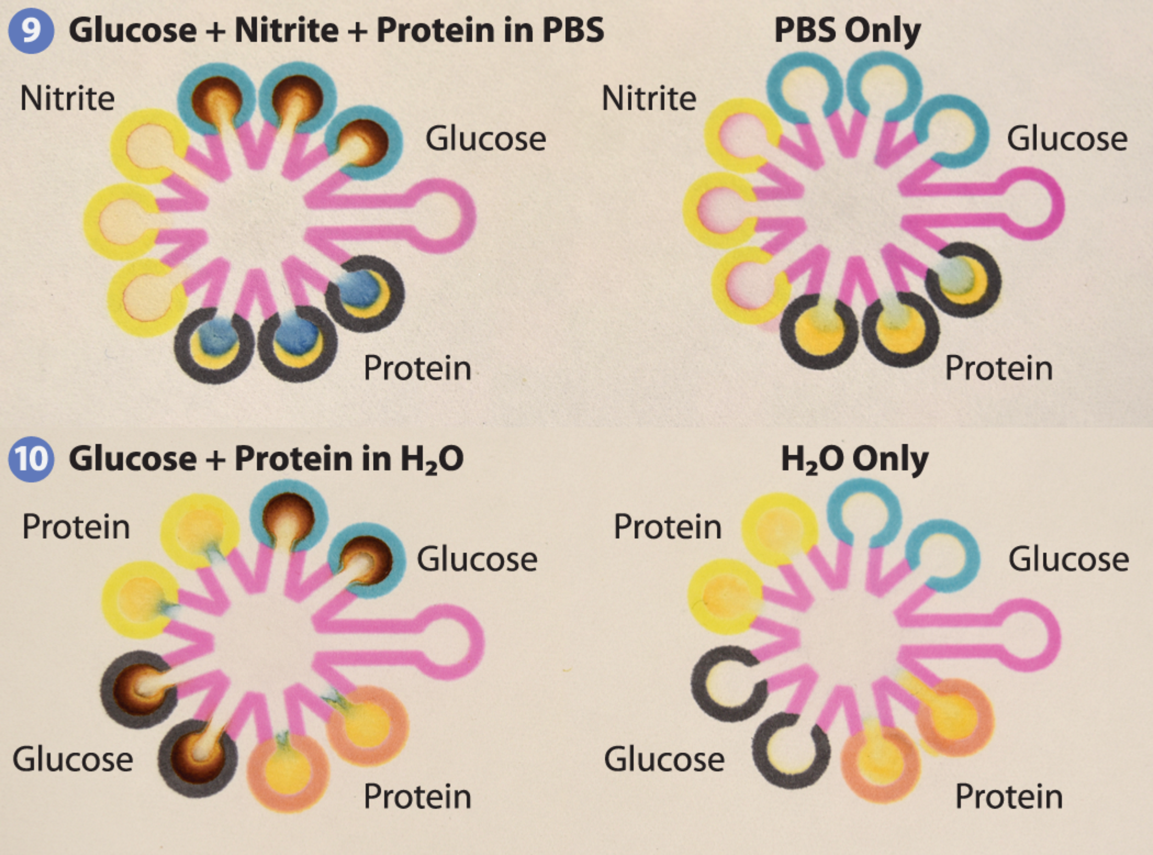


Figure 7.4: **Phases 9-10** Results from using the software-designed and -optimized paper microfluidic devices in a model urinalysis assay, detecting the presence (left sample) and absence (right sample) of glucose, nitrite, and protein in a 1x phosphate buffered saline (PBS) synthetic urine sample. While the glucose test sinks functioned as expected (with a color change from clear to dark brown only when glucose is present in the synthetic urine), the protein test exhibited at least some color change from yellow to blue in both the presence and absence of the protein BSA, and the nitrite test similarly displayed some color change from clear to red both with and without nitrites present. These issues were corrected during Phase 10 when testing modified version of the paper microfluidic device from Phase 9 updated by the software to now contain two different assays (glucose and protein) with four channels of each (in two sets of two). The device functions as expected, with color changes visible when the relevant analyte is present in the synthetic urine (left) and no color changes when the analyte is absent (right).

Chapter 8

Fabrication of multi-layer paper microfluidic devices

Most paper microfluidic devices consist of a single layer of paper that contains paths for fluid to travel via capillary action. To confine fluid to the intended paths, many devices use hydrophobic barriers created by printing wax-based inks using a standard office printer [11, 88, 22, 84]. This technique supports complex layouts of channels, reservoirs, and other features, but these layouts are confined to a single two-dimensional surface, and two fluid paths cannot cross each other. Recently demonstrated *multilayer* paper microfluidic devices can route fluids not only laterally in one layer, but also up and down to other layers [92, 31, 32, 73, 72, 114, 39, 87, 51, 60, 28]. This enables much more sophisticated operations than would be possible using single-layer devices. For example, Martinez *et al.* demonstrated multilayer paper fluidic multiplexers that allow the user to "program" the path followed by fluid through the device by pressing button-like features on the device [72].

A variety of methods for fabricating multilayer paper microfluidics have been proposed, as summarized in Table 8.1. However, these methods generally require specialized fabrication equipment or specialty chemicals, both of which limit the feasibility of these techniques in some settings. Many of these methods also incorporate additional materials into the paper microfluidic device, materials like polymer sheets and films [92], polymer treatments [31], wax [32], adhesive-backed tapes [73, 72, 114, 39, 87], and others. These materials can have unknown compatibilities with the fluids and reagents used in an assay, so when they come in contact with fluids during device operation, these materials could cause the assay to fail (or conversely, exposure to fluids could cause these materials to weaken or fail). Some multilayer paper fabrication techniques utilize spray-on adhesives to bond layers together[60, 51, 28], but this indiscriminate delivery of adhesives can adversely affect fluid flow in the device. Finally, combinations of both tape and adhesive have been proposed [39], but again these complex fabrication processes raise technical and practical challenges [69, 109].

In this work, a method for fabricating multilayer paper microfluidic devices is presented that requires no specialized equipment, chemicals, or techniques beyond those typically used to fabricate simpler, single-layer devices. The approach uses printed wax ink barriers as not only the channel walls that constrain fluid laterally, but also the "floor" and "ceiling" that constrain flow vertically. Since these wax barriers are already routinely used in paper microfluidics, their compatibility with various biological and chemical systems is already well established. The approach uses adhesives to hold the wax printed paper layers together and limits adhesive delivery to only the areas requiring bonding; this

Table 8.1: Summary of approaches to fabricating and assembling multi-layer paper microfluidic devices.

Materials	Equipment	Assembly	Comments
Paper, polylactic acid filaments, PTFE films, polyethylene terephthalate holder [92]	3D printer, heater	Paper cutouts in PET clamped together	Uses specialized equipment not readily available, inter-layer barriers require additional alignment with printed layers
Paper or polypropene film, wax, glass [32]	Laser cutter	Melt wax into paper sandwiched between two glass slides, cut channels, pump out excess wax	
Paper, photopolymers [31]	Polymer dispenser, laser	Dispense polymer into paper, activate with laser, bond layers with same process	
Paper, photoresist polymer, cellulose powder, adhesive tape[73, 74, 72]	Laser cutter, UV lamp, stencil	Pattern photoresist onto paper, expose to UV light, develop with acetone and isopropyl alcohol, punch holes in paper and tape, fill holes with cellulose powder, use double-sided tape between layers	
Paper, wax, adhesive tape polyester film, carbon ink[87]	Wax printer inkjet printer, hot plate	Print barriers, reheat wax to flow into paper, adhere single-sided adhesive tape to rear of paper, punch holes, print conductive pads and links on film, apply double-sided tape, fold and bake	
Fast-flow and absorbent papers, nylon membrane, adhesive tape, plastic laminate, wax ink [90, 20, 27]	Wax printer, plotter /cutter, heating element	Print barriers, reheat wax, cut and adhere double-sided adhesive tape between layers of paper and nylon membrane, laminate top and bottom of device for containment	
Paper, wax, spray adhesive, poly(3,4-ethylened ioxythiophene):polystyrene sulfonate multiwall carbon nanotubes (MWNTs)[39]	Wax printer, inkjet printer, heating method	Print barriers, reheat wax to flow into paper, print PEDOT:PSS and MWNTs, print and reheat overlying wax barriers, spray adhesive, fold into final device	Sprayed adhesive covers fluid channels impeding fluid flow
Paper, wax, uniform spray adhesive [114]	Wax printer, heat gun	Print barriers, reheat wax to flow into paper, punch holes, spray adhesive to entire paper, and adhere layers	
Paper, wax, uniform spray adhesive [60]	Wax printer, roller	Print pages, spray adhesive to one entire side of paper and adhere layers	
Paper, nylon, wax patterned spray adhesive[28]	Wax printer, oven, stencils	Print barrier designs on paper and nylon membrane, spray adhesive using stencils, adhere layers, bake	Patterned adhesives impact barrier efficacy
Paper, wax, patterned spray adhesive, tape [51]	Wax printer, hot plate, stencils	Print barriers, reheat wax to flow into paper, spray adhesive using stencils, fold, adhere layers, use single-sided tape to prevent leakage	
This work: Paper, wax, liquid adhesive	Wax printer, hot plate	Print barriers, reheat wax to flow into paper, punch holes, apply adhesive to barriers, adhere layers	No specialized equipment or chemicals

minimizes contact between the adhesive and fluids and does not adversely affect fluid flow inside the device. As a proof-of-concept, this fabrication method was used to create a multilayer microfluidic demultiplexer that allows the user to control fluid routing in three dimensions post-fabrication by simply pressing button-like locations on the device. This ten-layer paper demultiplexer functioned as intended, confirming that complex multilayer paper microfluidics can be fabricated quickly and easily using this method.

8.1 Materials and methods

8.1.1 Automated design of multilayer paper microfluidics

For this work, the software framework [86, 85] was used to generate different multilayer paper microfluidic device designs using components native to the framework's library. The ***well*** component consist of a circular or polygonal shape of some minimum border size enclosing an open disc of paper for fluid to wick into and/or through. ***Channel*** components are used to move fluids in a planar fashion and consist of a path of a desired width and length with hydrophobic borders of a minimum width on either side of the path. A ***floor*** can be created by placing a copy of the well or channel with a solid ink interior on a layer below the open component, thereby preventing fluid from seeping below the desired fluid zone. Similarly, a ***ceiling*** can be created using the same approach on a layer above the desired fluid zone, thereby preventing fluid from travelling upwards to another layer and preventing evaporation during operation. Finally, junction points defined as ***vias*** can connect various channels horizontally along the paper plane and vertically between layers of paper.

Three different multilayer paper microfluidic device designs were generated for this chapter:

- The first device used for testing consisted of multiple copies of a multi-level *via* for fluid to traverse multiple layers while still maintaining containment in both wax-bounded and adhesive-bounded layers.

- The second device (Fig 8.1) utilizes three *vias* connected by an open *channel* connecting a central reservoir *via* where the fluid is delivered and a distant destination *via* to which the fluid travels. A second closed *channel* connects the central reservoir and a distant destination *via*, following a fluid path that travels from the top layer of the device to the bottom, then a small distance laterally, and finally back up to the top.

- The third device is a radial demultiplexer (rMUX) we developed that was inspired by Martinez *et al.* [72] and allows for multiple fluid pathways to be chosen by an end-user post-fabrication. The rMUX (Fig 8.2) utilizes several *reservoirs*, *vias*, and closed *channels* to allow for path selection by pressing on the grey circular selection points; this forms a physical connection between layers and allows capillary action to draw the fluid upwards to the top reservoirs on the outer circle of the device. A large central *reservoir* connected to multiple *vias* serves as the source where the fluid is delivered, and fluid flows to one or more of the six destination *reservoirs* if the associated selection points are pressed.

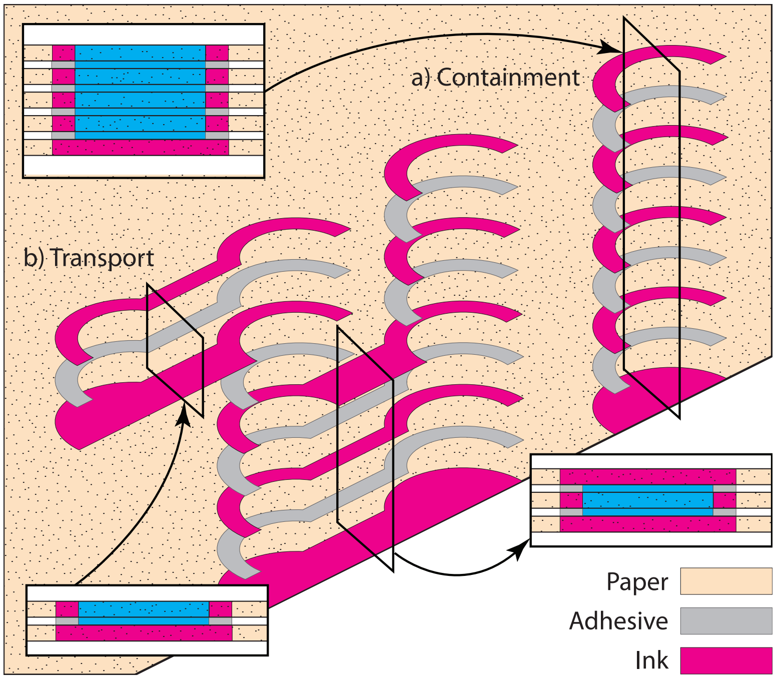


Figure 8.1: Exploded cross-sections of the containment test device **(a)** and the transport test device **(b)**. The containment device features a multi-layer reservoir for delivery of fluid to the device. The transport test device features a center reservoir for delivery of fluid to the device, an open transport channel to the left of the central reservoir and a closed channel to the right and bottom of the device. The solid hydrophobic magenta ink regions between reservoirs act as "walls" to confine fluids laterally, "floors" to prevent fluid from traveling downwards, and "ceilings" to restrict fluids from traveling upward. The adhesive, depicted in grey, is shown in the locations where it is delivered between pairs of layers.

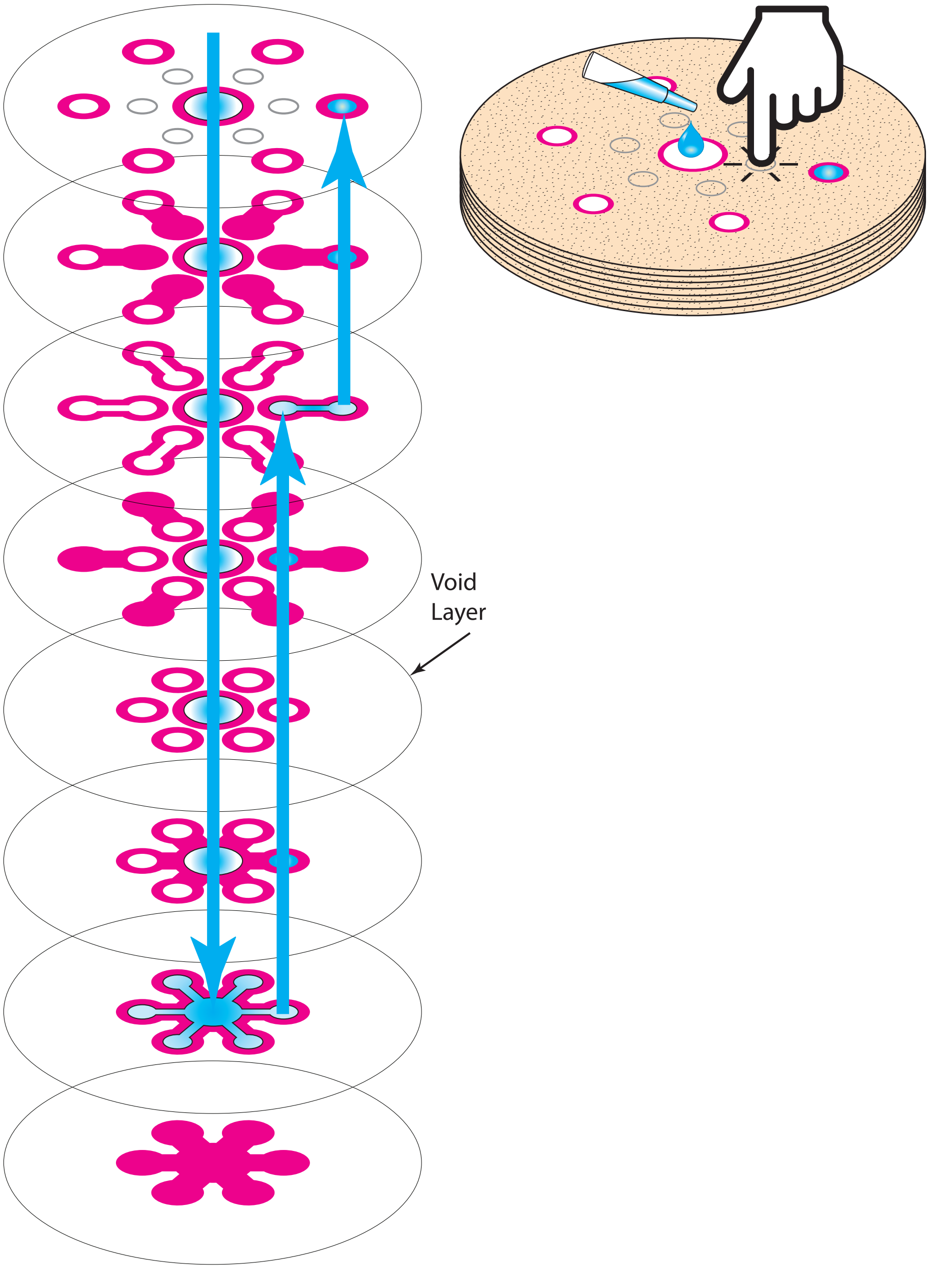


Figure 8.2: The radial paper microfluidic demultiplexer (rMUX) device design allows the user of the device to decide post-fabrication where to route fluids on the device. Before use, the device's void layers present gaps that prevent fluid from traveling upwards or downwards. When a user presses one or more of the grey-border selection points, the paper in those columns is compressed and capillary action is free to carry fluid up and down through the selected layers.

8.1.2 Containment and transport trials

The *containment trials* tested the ability of our paper microfluidic devices to contain liquid in a series of 5-layer vias (Fig 8.1a). Five sheets were chosen to be consistent with the number of sheets used in later trials involving fluid transport which required a minimum of 5 sheets to produce the necessary fluid pathways for testing.

The *transport trials* utilized a test design of our software framework's *channel* component that places a wax-filled "floor" channel below the fluid-carrying channel. The wax-filled channel acts as a fluid barrier to contain fluid in the neighboring fluid-filled channel, as shown in the left inset of Fig 8.1b. When a fluid-carrying channel is placed below another layer, a top wax-filled "ceiling" channel is placed in the layer above the fluid-carrying channels, thereby forming a "tunnel" for fluid travel within layers of the device as shown in the right inset of Fig 8.1b. Utilizing both multi-layer vias and channels, it is possible to route fluid both laterally and vertically between multiple layers using both gravity and capillary action to induce fluid flow.

8.1.3 Printing designs

The individual sheets in the multilayer paper microfluidic devices were fabricated using a standard process that utilizes Xerox ColorQube printers, a type of wax-ink-based office printer that has found widespread use in paper microfluidic device fabrication. These printers contain solid blocks of wax-based ink which the printer melts and deposits on the paper. Chandler *et al.* [12] showed that since this wax ink is hydrophobic, it can be used to make channel walls and define fluid paths in paper microfluidics. The print process uses four primary colors of printers: cyan (C), magenta (M), yellow (Y), and black (K)[58], which can be applied singly and in various combinations to produce full color images. The files output by our software are intended for use by these printers, although they could be adapted for use in other paper microfluidic fabrication techniques like computer-controlled cutting tools. Finally, since wax ink color can influence the behavior of fluids in paper microfluidic devices [84], each test device was printed in different colors: the containment test devices were printed in 15 different colors (the individual colors C, M, Y, and K, plus the combination colors CM, CY, CK, MY, MK, YK, CMY, CMK, CYK, MYK, and CMYK), and the transfer test devices were printed in a smaller subset of five different colors (C, M, Y, K, and CMYK).

8.1.4 Paper

The paper used was “101” fast qualitative filter paper (“Lab Nerd” brand; Amazon.com). To estimate the fluid capacity of the paper, we measured the thickness of a single sheet (100 μm) and assumed a 50% porosity, meaning that 50% of the paper volume could be occupied by fluid. The circular paper was trimmed to 200 × 123 mm rectangles for manual feeding into the printer. Once printed, each layout was held printed-side-up 5–7 cm above a 166 °C laboratory hotplate (temperature measured using an infrared thermometer) until wax melting was observed via color fading and blurring of printed edges. This step re-melts the wax ink so that it permeates the full thickness of the paper and creates an effective hydrophobic barrier.

8.1.5 Adhesives

Several classes of adhesives were chosen as candidates for assembling multi-layer paper devices. The criteria for adhesive selection included: 1) advertised hydrophobic nature (“waterproof”), 2) ability to bond both paper and wax printer ink, and 3) ability to bond materials without adversely affecting the device during or after fabrication. All seven adhesives chosen are readily available in home improvement centers. For each test device, the amount of adhesive used during fabrication was calculated by measuring the mass of the device before and after adhesive application.

- **Adhesive A** is a polyurethane base containing methyl ethyl ketone [40]; it was delivered via toothpicks to application areas.

- **Adhesive B** is a tetrachloroethylene-based adhesive [24] delivered to application areas using the manufacturer-provided nozzle.

- **Adhesive C** is an ethylene-vinyl acetate co-polymer and rosin ester adhesive [3] delivered by a standard hot-melt glue gun.

- **Adhesive D** is a non-toxic wood adhesive [29] that was applied using toothpicks.

- **Adhesive E** is a polyisocyanate prepolymer based on methylene diphenyl diisocyanate, polymeric diphenylmethane diisocyanate (pMDI), 4,4’-diphenylmethane diisocyanate, and diphenylmethane diisocyanate mixed isomers [97] and was delivered via toothpicks.

- **Adhesive F** is an ethyl cyanoacrylate adhesive in gel form [96], applied using the manufacturer-supplied nozzle.

- **Adhesive G** is a trade secret adhesive in aerosol form [1] that was delivered by spray nozzle to the entire mating side each sheet of paper at a distance of 18–20 cm and adhered following 30 seconds of drying time per manufacturer's instructions.

8.1.6 Aligning layers

The paper stock used in the experiments comes from the manufacturer as 200 mm diameter discs that must be cut down to rectangular shapes that the printer can use via the manual feed mechanism. Each sheet was cut down manually using a paper cutter, so the actual size of each sheet can vary somewhat. Consequently, each sheet of a multi-layer device must have alignment marks that are fixed relative to the elements on the printed page to guarantee accurate alignment independent of the edges of the sheet of paper.

Registration marks consisting of two lines crossing at right angles and surrounded by a circle were printed in the same locations on each sheet. Once printed, a hole centered in the registration mark can receive a registration pin to hold the sheets in alignment during adhesive application and device assembly. Two registration tables (Fig 8.3) were built with nails driven through the wood platforms at the appropriate locations to serve as the registration pins.

8.1.7 Test liquids

For testing the flow of liquids through the generated paper microfluidic devices, a dilute aqueous solution of 2% (by volume) blue food coloring was delivered to the source reservoir of each device using a pipette. Our software framework calculated a device volume

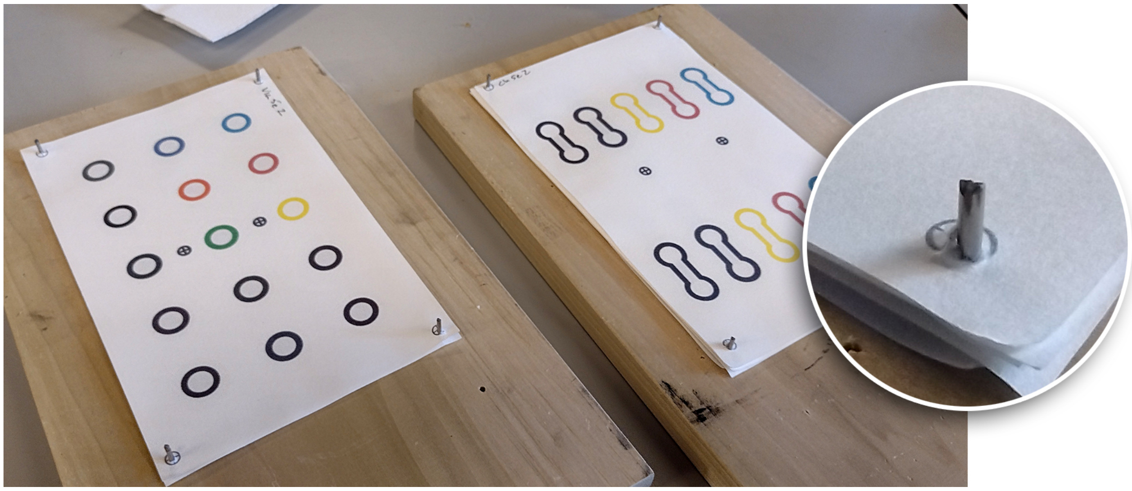


Figure 8.3: Aligning multiple paper layers using printed registration marks and nails as registration pins (inset).

of 30.7 μL for each of the containment trials and 100.85 μL for each of the transport trials. As a "stress test" for the devices, excessively-large fluid volumes of 150 μL (for the containment trials) and 200 μL (for the transport trials) were delivered in an attempt to induce failures in the devices' wax barriers or adhesive bonds. For each adhesive, the containment devices were filled first and the transport devices were filled second. Each test run was photographed from initial fluid delivery to completion and/or failure of the device due to either drying or barrier failure. The total elapsed operating time of each device was defined as the time between fluid delivery and the arrival of fluid at the destination. Upon completion, any remaining fluid was removed by pipette, and each device was photographed a final time.

8.2 Results and analysis

8.2.1 Aligning layers

It was observed that the paper stock was susceptible to tearing during the alignment phase, and this led to one device with an alignment error (Fig 8.4) that was not initially evident but became apparent upon inspection after device testing.

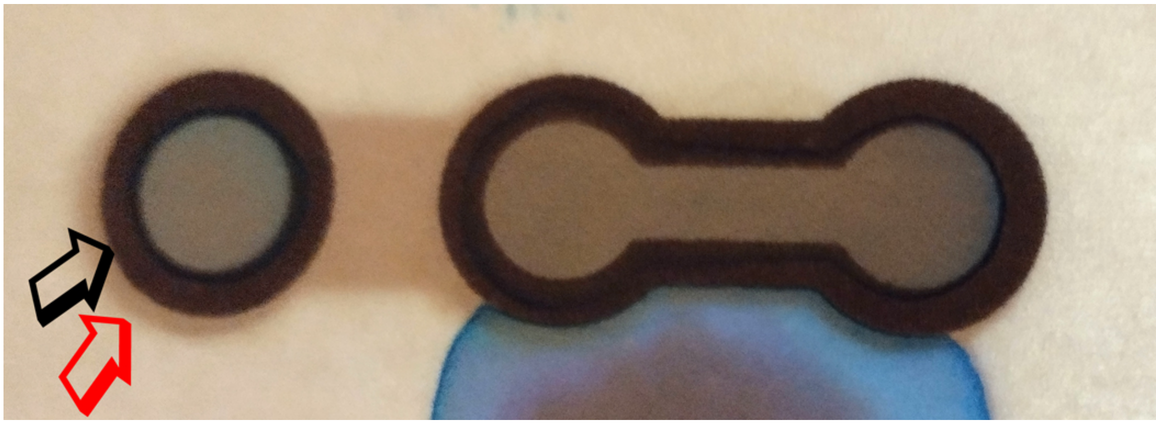


Figure 8.4: In this multilayer paper microfluidic device fabricated using adhesive A, one layer was misaligned during assembly, which led to a loss of fluid containment between the layers. The red and black arrows indicate the shift from proper alignment.

8.2.2 Containment trials

To test the ability of the multilayer microfluidic devices to control fluid flow in the vertical dimension (between paper layers), copies of the containment test device were fabricated using the seven adhesives and 15 wax ink color combinations and subjected each device to testing. Since different combinations of wax ink colors have been shown to have different behaviors as fluid barriers [84], the full range of color combinations was included to determine whether the adhesives would impact the performance of those barriers.

- **Adhesive A** had consistent adhesive delivery across layers due to the manual toothpick-based application, but some device sections had premature bonding due to the adhesive's short flash time (the amount of time that passes before the adhesive begins to cure). Containment devices fabricated using adhesive A had the second-most fluid-flow failures in our testing (5 failures). However, most of those failures occurred in devices containing C, Y, and CMK wax ink colors (see Tables 8.3 and 8.4); these ink colors were found to be less effective as barriers in previous work [84], so the failures observed here may also be due to ink color, not just the adhesive.

- **Adhesive B**'s wide delivery nozzle and hand-squeezed pumping led to inconsistent and excessive adhesive delivery across layers. Containment devices fabricated using adhesive B had the most failures in our testing (7 failures). As with adhesive A, most of the failures in adhesive B devices occurred in devices using certain color combinations that were known to form weak barriers in previous work [84], so again these failures may not be solely due to the adhesive.

- **Adhesive C** was delivered using a hot glue gun with a pressure-sensitive trigger system, which led to over-application and gaps between sheets due to the thickness of the adhesive once cured (Fig 8.5).

- **Adhesive D** had the longest work time before drying, which provided greater control of adhesive location and volume during toothpick-based application.

- **Adhesive E**'s high viscosity made it difficult to regulate the amount applied through the manufacturer-provided nozzle. Additionally, the adhesive tended to spread after device bonding and adversely affected fluid travel inside the device (Figs 8.6 and 8.7). The adhesive also failed to maintain full adhesion after curing. In spite of these deficiencies, containment devices fabricated using adhesive E successfully contained the fluid across all layers and for each wax ink color.

- **Adhesive F** was delivered using the manufacturer-provided nozzle. After application, the adhesive seeped beyond the target application area and rendered the entire surface of each device hydrophobic (Figs 8.8 and 8.9).

- **Adhesive G** was delivered by aerosol spray. Consequently, the adhesive was not confined to the bonding areas; adhesive droplets also covered the fluid channels and reservoirs where they rendered these features partly hydrophobic. Previous works [51, 60, 28] using this adhesive also mention this effect but still reported successful device operation (albeit with increased fluid travel times).

The combined results from the containment trials are shown in Fig 8.10 and summarized in Tables 8.2–8.4.

8.2.3 Transport trials

Having tested the ability of the containment test devices to control fluids during vertical travel in the previous section, the transport test devices' ability to simultaneously

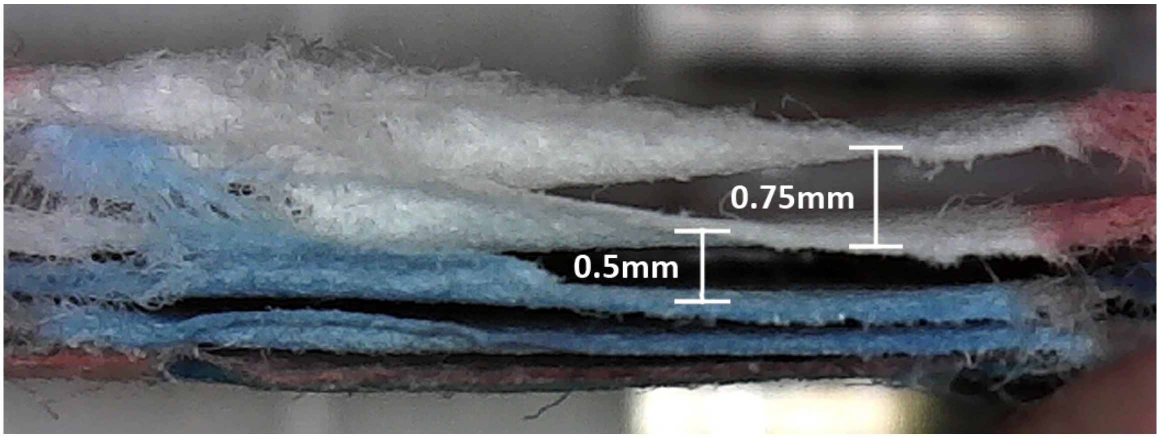


Figure 8.5: A high-volume nozzle and a difficult-to-regulate delivery system led to excessive amounts of adhesive C being delivered while assembling this device; this created air-filled gaps between layers that inhibited vertical fluid travel. In this cross-sectional side view of a containment trial, blue fluid has traveled upward through the bottom two layers, but air gaps have stopped the fluid from traveling into the top two layers.

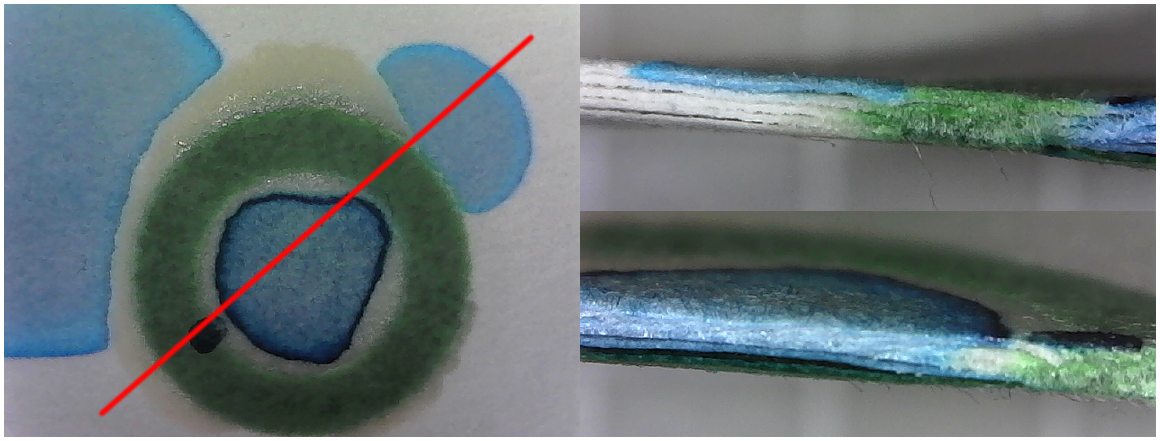


Figure 8.6: (Left) A multilayer paper containment trial device fabricated using adhesive E. When cut along the red line, the resulting cross sections show that while the adhesive spread somewhat and constricted the blue fluid, the fluid was nonetheless successfully confined (bottom right). The fluid present only on the top layer (upper right) is from a manual pipetting error while loading the device and not a barrier failure.

control both vertical and horizontal fluid travel was next explored. The results from fabricating and testing transport devices using each of the seven adhesives and four wax ink color combinations are shown in Fig 8.11 and summarized in Tables 8.5 and 8.6.

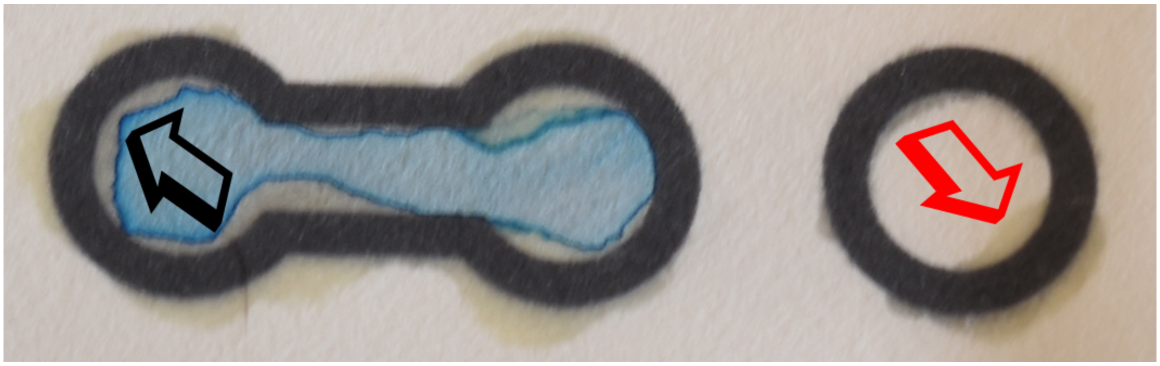


Figure 8.7: Inconsistent delivery of adhesive E led to adhesive spreading into the fluid-containing channels of the devices (arrows); this constricts the fluid path (blue) and reduces the fluid capacity of the device.

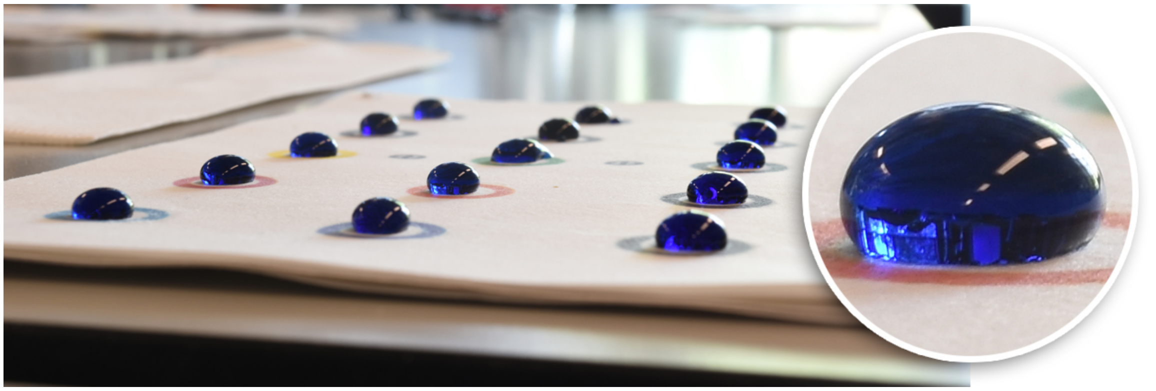


Figure 8.8: Adhesive F travelled well beyond its application areas and rendered the entire paper layer hydrophobic, as evidenced by these blue droplets that are unable to wet the paper beneath them.

- Devices fabricated using **adhesives B and E** showed the best performance for both containment and transport. Adhesive B devices successful transport for all wax ink colors and only 2 containment failures in the open channel and 1 fluid containment failure in the closed channel. Adhesive E devices had 1 containment failure in the magenta closed channel and 1 transport failure in the CMYK closed channel.

- Due to **adhesive F**'s tendency to render the entire sheet hydrophobic (Figs 8.8–8.9), no fluid penetrated the top layer of the adhesive F transport devices.

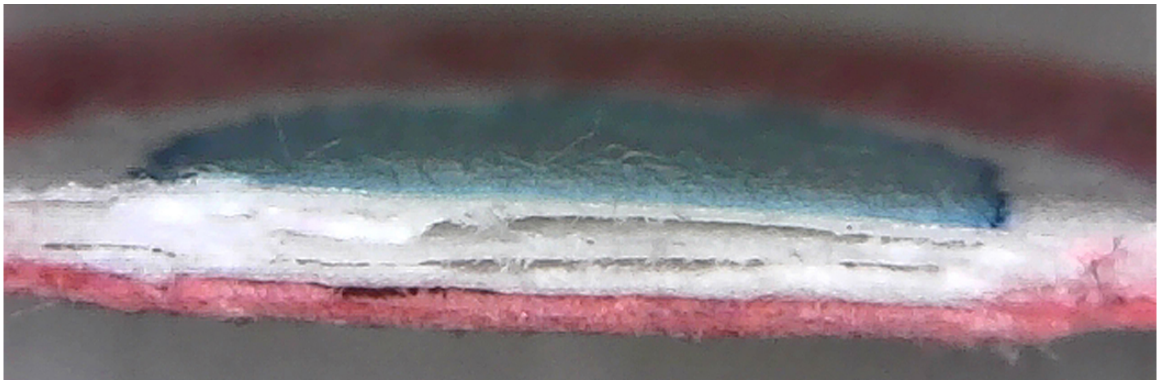


Figure 8.9: A cross-sectional side view of a multilayer paper device fabricated using adhesive F reveals that the blue fluid stained the top layer slightly but did not penetrate into any of the layers.

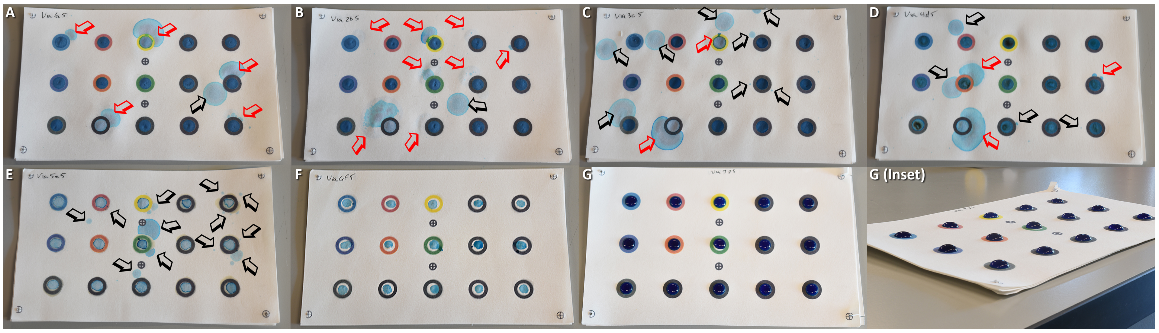


Figure 8.10: Combined results from fluid containment testing of multilayer paper microfluidic devices fabricated using each of the seven adhesives (A-G) and 15 wax ink color combinations. Black arrows indicate manual pipetting errors and red arrows indicate actual barrier or adhesive failures. The oblique view of the devices fabricated using adhesive G (inset) show that the adhesive G devices (like the adhesive F devices) exhibited unwanted hydrophobicity that limited the flow of fluid into the lower device layers.

- **Adhesive D** was certainly the worst performing adhesive overall, with 10 containment failures and 7 transport failures occurring in all wax ink color variations under examination.

- For the **open-channel devices**, fluids generally traveled to the destination reservoirs successfully in devices fabricated using adhesives A, B, D, and E. Devices fabricated using Adhesive C failed due to the air gaps (Fig 8.5) created by excess adhesive.

Table 8.2: Mass of adhesive applied between each pair of paper layers (from bottom layer 1 to top layer 5) and total mass of adhesive per device, for each of the seven adhesives (A–G) tested in this work, for both containment and transport experiments.

Containment trials

Adhesive	Between layers 1 and 2	Between layers 2 and 3	Between layers 3 and 4	Between layers 4 and 5	Total adhesive mass
A)	0.144 g	0.127 g	0.253 g	0.109 g	0.632 g
B)	0.464 g	0.475 g	0.330 g	0.205 g	1.474 g
C)	0.629 g	0.980 g	1.065 g	0.913 g	3.587 g
D)	0.477 g	0.098 g	0.161 g	0.121 g	0.856 g
E)	0.462 g	0.345 g	0.192 g	0.137 g	1.136 g
F)	0.221 g	0.207 g	0.339 g	0.197 g	0.965 g
G)	0.354 g	0.205 g	0.243 g	0.331 g	1.133 g

Transport trials

Adhesive	Between layers 1 and 2	Between layers 2 and 3	Between layers 3 and 4	Between layers 4 and 5	Total adhesive mass
A)	0.185 g	0.110 g	0.085 g	0.197 g	0.577 g
B)	0.531 g	0.318 g	0.415 g	0.715 g	1.978 g
C)	1.278 g	1.188 g	1.207 g	2.368 g	6.041 g
D)	0.193 g	0.186 g	0.141 g	0.300 g	0.820 g
E)	0.219 g	0.327 g	0.259 g	0.378 g	1.182 g
F)	0.230 g	0.231 g	0.411 g	0.522 g	1.393 g
G)	0.332 g	0.210 g	0.333 g	0.229 g	1.104 g

- Among the **closed channel devices**, all adhesives performed well across most wax ink colors. Indeed, even the otherwise-poor-performing adhesive D yielded functional devices when the flow channels are sandwiched between magenta wax ink layers, as shown in Fig 8.12. This shows that wax-ink-surrounded fluid channels are highly reliable for transporting fluids in multilayer devices.

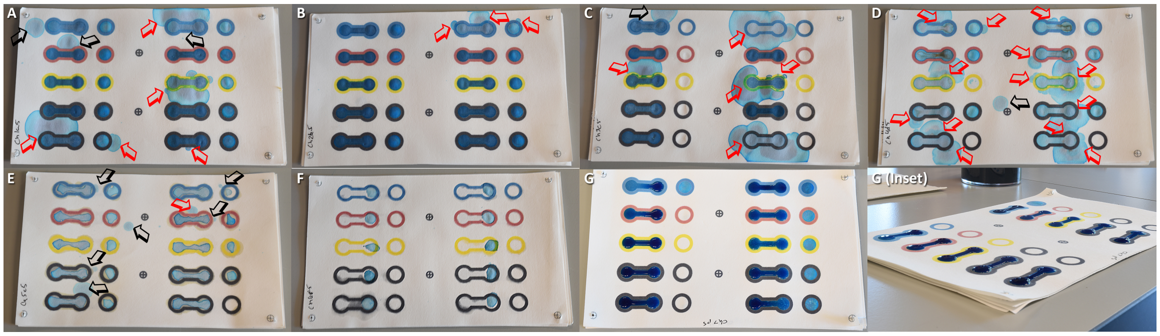


Figure 8.11: Results from fluid transport testing of multilayer paper microfluidic devices fabricated using each of the seven adhesives (A–G) and five selected wax ink color combinations (C, M, Y, K, and CMYK). Black arrows indicate manual pipetting errors and red arrows indicates actual barrier or adhesive failures. The oblique view of the devices fabricated using adhesive G (inset) again demonstrate the unwanted hydrophobicity that limited the flow of fluid into the lower device layers in devices that used adhesive G.

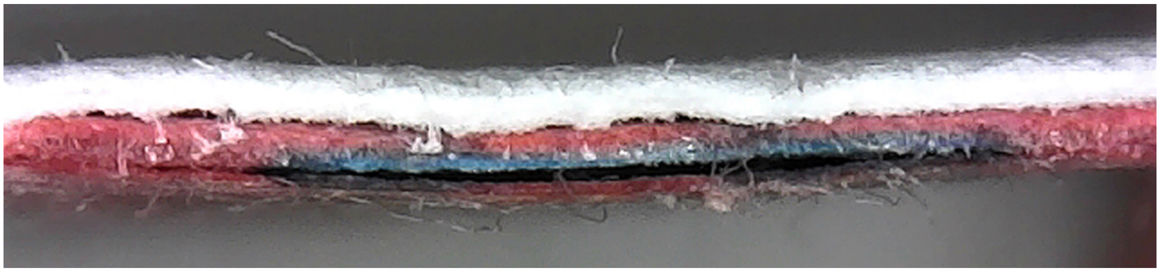


Figure 8.12: The closed channel design which uses printed wax-ink barriers (magenta) above and below the fluid transport channel (blue) is highly effective at maintaining flow in the channel while blocking flow to neighboring layers.

8.2.4 Radial demultiplexer (rMUX)

Based on the results from the containment and transport trials, it decided that *adhesive B* and *magenta wax ink* yielded the best-performing multilayer paper microfluidic devices. Adhesive B and magenta ink was then used to fabricate several rMUX devices of the design shown in Fig 8.2. Holes of 5 mm diameter were punched in the vias on layer 4 to create the necessary air gaps that restrict fluid travel until compressed at runtime by the end user. A single punched sheet of paper provides an air gap of about 0.1 mm; larger air gaps were obtained by stacking multiple sheets of punched paper. For more accurate glue delivery during fabrication, adhesive B's manufacturer-provided nozzle was modified by attaching a trimmed plastic pipette tip. Finally, the selected channel pattern for each rMUX was assigned at random after fabrication and before fluid delivery.

The first set of six rMUX test devices (Trial 1; Fig 8.13) were fabricated using a 0.1 mm thick air gap (1 sheet of punched paper) and a total of 8 device layers. After selecting the desired fluid paths by pressing a black ballpoint pen into the selection points, 170 μL of blue fluid was delivered to the center of each device. The total time elapsed for all the trials was approximately 18 minutes, with 7 minutes being the minimum amount of time before the first destination reservoir was filled. The results in Fig 8.13 show that fluid still flowed to several non-selected reservoirs, suggesting that the 0.1 mm thick air gap was not large enough to stop fluid flow. Additionally, some selected reservoirs filled only partially, indicating that the delivered volume of fluid is too small.

For rMUX Trial 2 (Fig 8.14), devices were fabricated using a 0.2 mm air gap (two sheets of punched paper) and a total of 9 device layers. The volume of fluid delivered to

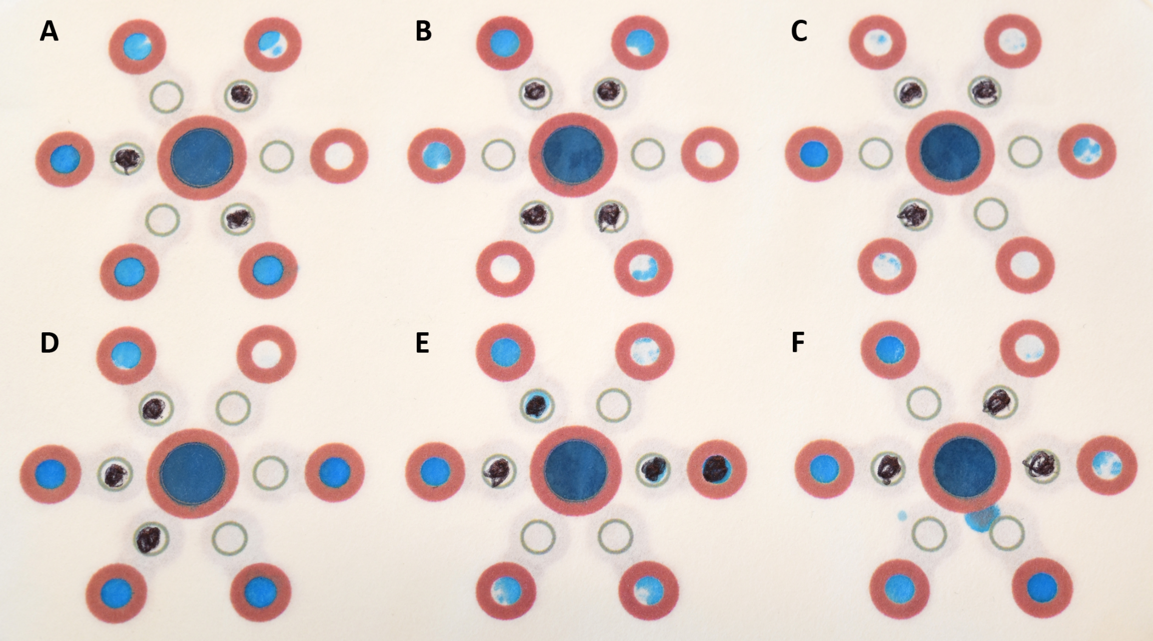


Figure 8.13: Trial 1: Results from testing radial demultiplexer (rMUX) devices fabricated using adhesive B, magenta wax ink barriers, and 0.1 mm thick air gaps at the selection points. In each rMUX test, a black ballpoint pen was used to press the selection points and select the desired fluid paths (dark black ink spots mark the selected paths), then blue fluid was added to the central reservoir. The results show that a 0.1 mm air gap is not wide enough to stop fluid travel to non-selected reservoirs.

the device was was also increased to 180 μL for devices A, C, D, E, and F; and 220 μL for device B. The total elapsed time for Trial 2 increased to over 30 minutes, with the first destination reservoir filling 5 minutes into the run. The results in Fig 8.14 show that fluid flowed successfully to most of the selected reservoirs. However, some fluid flow to non-selected reservoirs still occurred, indicating that the air gap at the selection points is still too small.

For rMUX Trial 3 (Fig 8.15), devices were fabricated using a 0.3 mm air gap (three sheets of punched paper) and a total of 10 device layers, and delivered fluid volume was again increased to 200 μL for devices A, D, E, and F; 225 μL for device C; and 300 μL

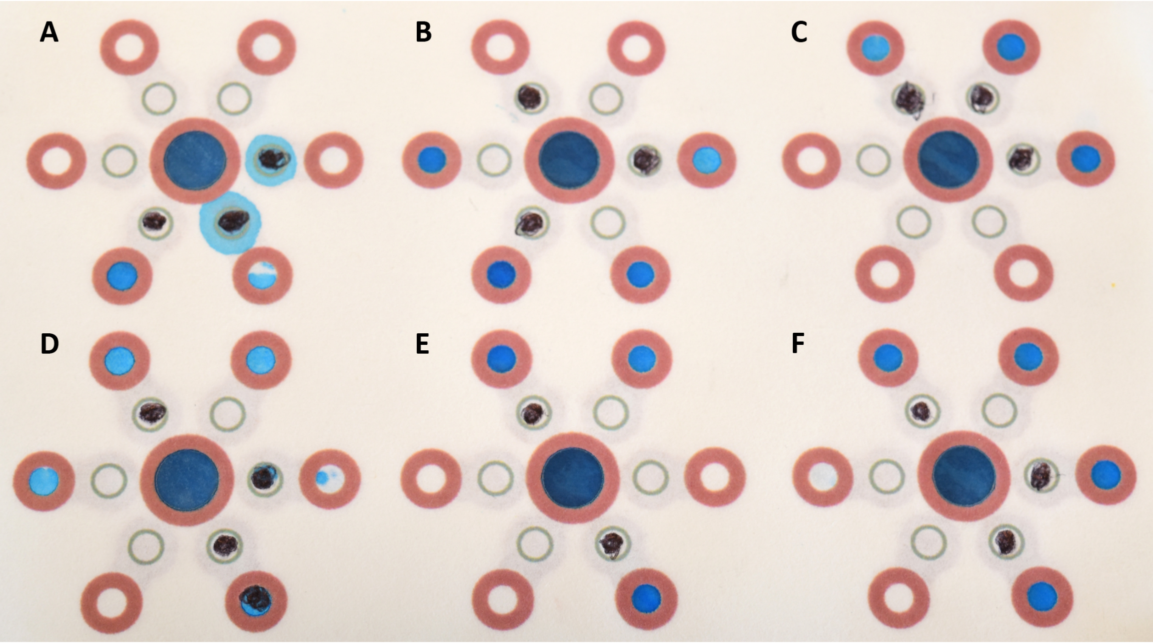


Figure 8.14: Trial 2: Results from testing rMUX devices with 0.2 mm thick air gaps at the selection points and increased volumes of fluid delivered to the central reservoirs. In general, fluid reached the selected destination reservoirs, though fluid also still reached some non-selected reservoirs. Fluid leakage at the selection points in device A was attributed to damage caused by excessive pressure applied by the ballpoint pen during selection.

for device B. The total elapsed time for Trial 3 increased to over 30 minutes, with the first destination reservoir filling in 6.5 minutes. The results in Fig 8.15 show that no fluid flowed to non-selected reservoirs, indicating that the 0.3 mm air gap is thick enough for reliable function. In addition, fluid successfully reached all but one of the selected reservoirs; the single failure (device C) was again attributed to damage caused by excessive pressure applied by the ballpoint pen during selection. In future work, this damage might be avoided by pressing the selection points using a more-blunt tool.

Finally, it was observed that the required air gap thickness for proper rMUX function, 0.3 mm, is comparable to the distance between layers observed in devices fabricated using adhesive C (Fig 8.5). This suggests that by using adhesive C instead of adhesive B, an adequate air gap could be obtained using adhesive alone, thereby further reducing the amount of materials required to fabricate user-selectable paper microfluidic devices like the ones studied here.

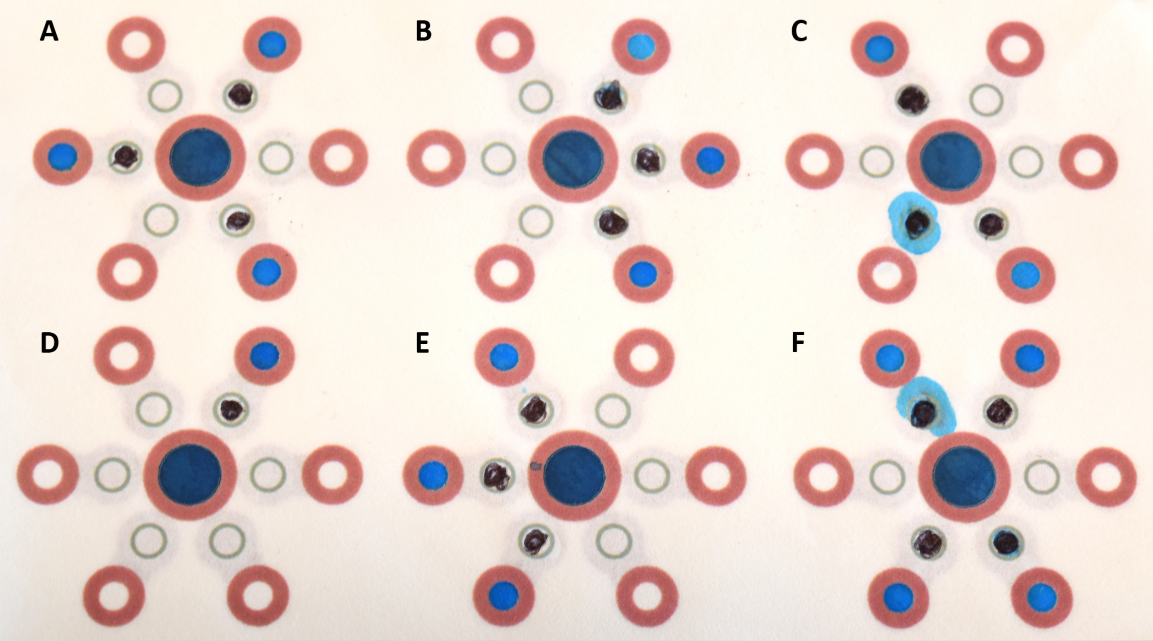


Figure 8.15: Trial 3: Results from testing rMUX devices with 0.3 mm thick air gaps at the selection points and increased volumes of fluids delivered to the central reservoirs. The devices operated as intended; no fluid flowed to the non-selected reservoirs, and fluid *did* flow to all selected reservoirs except one (device C, which failed due to pressure-induced damage at the selection point).

Table 8.3: Results from testing containment devices for each adhesive (A–G) and each combination of wax ink colors (C, M, Y, K, MK, CM, MY, CY, YK, CK, CMY, CMK, CYK, MYK, and CMYK). For each experiment, the left dot indicates the fluid containment status immediately after fluid reaches the destination, and the right dot represents the fluid containment status after additional time has passed. A black circle (●) indicates that the device successfully contained the fluid, and a red circle (🔴) indicates that the device failed to contain the fluid.

Adhesive			Colors		
A)	● 🔴 C	● ● M	🔴 🔴 Y	● ● K	● ● MK
	● ● CM	● ● MY	● ● CY	● ● YK	● ● CK
	● ● CMY	🔴 🔴 CMK	● ● CYK	● ● MYK	● 🔴 CMYK
B)	● 🔴 C	● ● M	🔴 🔴 Y	● 🔴 K	● 🔴 MK
	● ● CM	● 🔴 MY	● ● CY	● 🔴 YK	● ● CK
	● ● CMY	🔴 🔴 CMK	● 🔴 CYK	● ● MYK	● ● CMYK
C)	● ● C	● ● M	● 🔴 Y	● ● K	● ● MK
	● ● CM	● ● MY	● ● CY	● ● YK	● ● CK
	● ● CMY	🔴 🔴 CMK	● ● CYK	● ● MYK	● ● CMYK
D)	● ● C	● ● M	● ● Y	● ● K	● ● MK
	● ● CM	🔴 🔴 MY	● ● CY	● ● YK	● 🔴CK
	● ● CMY	🔴 🔴 CMK	● ● CYK	● ● MYK	● ● CMYK
E)	● ● C	● ● M	● ● Y	● ● K	● ● MK
	● ● CM	● ● MY	🔴 🔴 CY	● ● YK	● ● CK
	● ● CMY	● ● CMK	● ● CYK	● ● MYK	● ● CMYK
F)	● ● C	● ● M	● ● Y	● ● K	● ● MK
	● ● CM	● ● MY	● ● CY	● ● YK	● ● CK
	● ● CMY	● ● CMK	● ● CYK	● ● MYK	● ● CMYK
G)	● ● C	● ● M	● ● Y	● ● K	● ● MK
	● ● CM	● ● MY	● ● CY	● ● YK	● ● CK
	● ● CMY	● ● CMK	● ● CYK	● ● MYK	● ● CMYK

Table 8.4: Fluid containment failures in the containment test devices, organized by adhesive type (columns A–G) and wax ink color (rows C–CMYK).

Wax ink				Adhesives				
color	A	B	C	D	E	F	G	Total
C	1	1	-	-	-	-	-	2
M	-	-	-	-	-	-	-	0
Y	1	1	1	-	-	-	-	3
K	-	1	-	-	-	-	-	1
MK	-	1	-	-	-	-	-	1
CM	-	-	-	-	-	-	-	0
MY	-	-	-	1	-	-	-	1
CY	-	1	-	-	-	-	-	1
YK	-	1	-	-	-	-	-	1
CK	1	-	-	1	-	-	-	2
CMY	-	-	-	-	-	-	-	-
CMK	1	1	1	1	-	-	-	4
CYK	-	-	-	-	-	-	-	-
MYK	-	-	-	-	-	-	-	-
CMYK	1	-	-	-	-	-	-	1
Total	5	7	2	3	0	0	0	17

Table 8.5: Results from testing the transport devices for each adhesive (A–G) and each of five selected combinations of wax ink colors (C, M, Y, K, and CMYK) shown in Fig 8.11. Two copies of each device were tested, designated left (*L*) and right (*R*). For each experiment, the first and third dots indicates the *fill* status of the device's open and closed channels, respectively; a filled black symbol (●) indicates successful filling of the channel reservoir and completion of the fluid travel, and an empty red symbol (⭕) indicates the reservoir was not successfully filled (and therefore the failure failed to route the fluid correctly). The second and fourth dots indicate the *containment* status of the open and closed channels, respectively; an empty black symbol (○) indicates successful fluid containment, and a filled red symbol (🔴) indicates a fluid containment failure.

Trial	Wax ink colors				
A) L	● ○ ● ○ C	● ○ ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● 🔴 ● 🔴 CMYK
A) R	● 🔴 ● ○ C	● ○ ● ○ M	● 🔴 ● ○ Y	● ○ ● ○ K	● 🔴 ● ○ CMYK
B) L	● ○ ● ○ C	● ○ ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● ○ ● ○ CMYK
B) R	● 🔴 ● 🔴 C	● ○ ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● ○ ● ○ CMYK
C) L	● ○ ⭕ ○ C	● ○ ⭕ ○ M	● 🔴 ⭕ ○ Y	● ○ ⭕ ○ K	● ○ ⭕ ○ CMYK
C) R	● 🔴 ⭕ ○ C	● ○ ⭕ ○ M	● 🔴 ⭕ ○ Y	● ○ ⭕ ○ K	● 🔴 ⭕ ○ CMYK
D) L	● 🔴 ● 🔴 C	● ○ ● ○ M	● 🔴 ⭕ ○ Y	● 🔴 ● ○ K	● 🔴 ⭕ 🔴 CMYK
D) R	● 🔴 ● ○ C	● 🔴 ● ○ M	● 🔴 ⭕ 🔴 Y	● 🔴 ● ○ K	● 🔴 ⭕ 🔴 CMYK
E) L	● ○ ● ○ C	● ○ ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● ○ ● ○ CMYK
E) R	● ○ ⭕ ○ C	● 🔴 ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● ○ ⭕ ○ CMYK
F) L	⭕ ○ ⭕ ○ C	⭕ ○ ⭕ ○ M	⭕ ○ ⭕ ○ Y	⭕ ○ ⭕ ○ K	⭕ ○ ⭕ ○ CMYK
F) R	⭕ ○ ⭕ ○ C	⭕ ○ ⭕ ○ M	⭕ ○ ⭕ ○ Y	⭕ ○ ⭕ ○ K	⭕ ○ ⭕ ○ CMYK
G) L	● ○ ● ○ C	● ○ ⭕ ○ M	● ○ ⭕ ○ Y	● ○ ⭕ ○ K	● ○ ⭕ ○ CMYK
G) R	● ○ ● ○ C	● ○ ● ○ M	● ○ ● ○ Y	● ○ ● ○ K	● ○ ● ○ CMYK

Table 8.6: Fluid containment and transport failures in the transport test devices in Fig 8.11, organized by adhesive type (columns A–G) and wax ink color (rows C, M, Y, K, and CMYK). The failures are categorized by whether the failure occurred in the *open channel* to the left of the central reservoir or the *closed channel* on the right. Additionally, failures are further categorized as *containment failures* (fluid escaped intended bounds) or *completion failures* (fluid did not reach destination).

Color	Failure type	A	B	C	D	E	F	G	Total
C	Open containment	-	-	-	2	-	2	-	4
	Open completion	1	2	-	-	-	-	-	3
	Closed containment	-	-	2	1	1	2	-	6
	Closed completion	-	1	1	-	-	-	-	3
M	Open containment	-	-	-	-	-	2	-	3
	Open completion	-	-	-	1	1	-	-	3
	Closed containment	-	-	2	-	-	2	1	5
	Closed completion	-	-	-	-	-	-	-	0
Y	Open containment	-	-	-	-	-	2	-	2
	Open completion	1	-	2	2	-	-	-	5
	Closed containment	-	-	2	2	-	2	1	7
	Closed completion	-	-	1	1	-	-	-	2
K	Open containment	-	-	-	-	-	2	-	2
	Open completion	-	-	-	2	-	-	-	2
	Closed containment	-	-	2	-	-	2	1	5
	Closed completion	-	-	-	-	-	-	-	0
CMYK	Open containment	-	-	-	-	-	2	-	2
	Open completion	1	-	1	2	-	-	-	4
	Closed containment	-	-	2	2	1	2	1	8
	Closed completion	2	-	-	2	-	-	-	4
Total		5	3	15	17	3	20	4	67

Part IV

Conclusion

Chapter 9

Conclusions

Paper-based microfluidics provide portable laboratory tests without expensive actuation equipment. Paper microfluidics replaces externally-actuated pumps and high-voltage-driven electrodes with low-cost wicking materials, thereby reducing the cost of fabricating new devices and performing diagnostic tests; however, this does not automatically imply a reduction in complexity of the design process.

To address these concerns, this work has introduced a software framework that can simplify and automate the design process. Developers can use the framework to prototype and test new designs, which includes varying the underlying substrates and experimenting with passive flow networks. The framework also provides the capability to reliably reproduce devices streamlined for in-situ fabrication and to provide tools for testing and analysis of the designs; this, in turn, informs the alteration and rapid prototyping of design variations, helping to account for test results, environmental conditions, impact on physical substrates and fluids, and accuracy under test.

In addition to generating device designs, the software framework also calculates the volume of fluid contained by each design based on the component used and the paper characteristic. Using this information, a researcher can find an optimal design for a certain sample size without having to fabricate and test the device. Currently, the software calculates device volume using a straightforward geometric calculation based on the planar area of the generated device multiplied by the thickness and pore density of the paper. However, the framework will also support user-defined functions for device volume calculations in future versions.

The software framework can still be useful even after a researcher has generated a satisfactory device design for a given application. For example, substituting one type of paper for another could change the wicking behavior or fluid capacity of the paper, requiring re-optimization of some device geometry parameters. Likewise, incorporating a new assay chemistry into an existing device may require generating new device designs with different parameters. Finally, a paper microfluidic device designed to work in one location (perhaps, a climate-controlled laboratory) might not function as intended in a very different location with different temperature, humidity, etc., so aspects of the device design would need to be optimized again. In these and other scenarios, using the software framework would expedite the necessary design modifications.

The nD-Tree was introduced along with associated algorithms to perform efficient component placement during physical design of multi-layer paper-based microfluidic devices. The nD-Tree addresses bounding and conflicting conditions that may occur when placing components that span multiple device layers. After having integrated the ND-Tree into a paper microfluidic device design framework, future work will investigate placement and routing algorithms for multi-layer paper microfluidic devices leveraging the ND-Tree to perform collision detection to ensure legality of the result.

The Radar approach to placement and routing demonstrates that layouts that can be generated and optimized for metrics such as fluid travel and usage, device area, and total materials required – all desirable goals to help meet ASSURED criteria.

The results shown in this work demonstrate that designers of paper microfluidic devices can use different ColorQube ink colors to impart specific behaviors to their devices. Among the single-color inks, it was observed that magenta had the fewest barrier failures, a finding that is consistent with previous work [93]. It was also found that certain ink color combinations perform even better than magenta: the combinations CY, MK, YK, CMY, CYK and MYK had no barrier failures in our study.

Therefore, for applications requiring the strongest possible hydrophobic barrier, researchers are advised to use these combinations. Based on these findings that ink mass correlates with barrier effectiveness (except for K), one can easily create same-size printed samples of all 15 ink color combinations, weigh them, and determine which non-K color combinations result in the largest amount of ink delivered to the paper (and are therefore most likely to provide the strongest barrier to fluid flow).

Findings show that ink color selections can be used for more than just reducing barrier failures. By selecting a color with a known propensity to fail in a certain manner, a paper microfluidic "antifuse" was created that can serve as a fail-safe protection against operator error in lateral flow devices.

A method was demonstrated for fabricating complex multilayer paper microfluidic devices that requires no specialized equipment or chemicals, minimizes contact between fluids and materials with unknown compatibilities, and has no observed adverse affect on fluid flow. Using this method, paper microfluidic demultiplexers were successfully fabricated and tested that let the user "program" the device operation by pressing pushbutton-like structures on the device, but far more complex devices can also be fabricated using this technique. By eliminating many of the practical barriers to fabricating multilayer paper microfluidics, this approach can accelerate the development of sophisticated paper microfluidic tools for a wide range of important applications.

Finally, while this demonstration was limited in various ways for brevity's sake, the software framework is extensible to support many other use cases. For example, while limiting to radial device designs with a single sample and multiple assays, the software framework allows users to define custom device designs. Additionally, while the device designs generated in this study were all single layer, the framework is capable of generating multi-layer paper microfluidic devices as well. Lastly, while we utilized a wax-ink-based printer to fabricate devices in this study, the software is also capable of generating designs suitable for other fabrication tools such as computer-controlled paper cutters.

Bibliography

[1] 3M. 3m^{TM} super 77^{TM} multipurpose adhesive. https://multimedia.3m.com/mws/mediawebserver?mwsId=SSSSSuUn_zu8l00x482xN8mvNv70k17zHvu9lxtD7SSSSSS--.

[2] Arash Abadian and Shahin Jafarabadi-Ashtiani. Paper-based digital microfluidics. *Microfluidics and Nanofluidics*, 16(5):989–995, 2014.

[3] Adhesive Technologies, Inc. Material safety data sheet item : Ad-tech 220. https://cdn3.evostore.io/documents/vow/sb05212_coshh.pdf.

[4] Nada Amin, William Thies, and Saman Amarasinghe. Computer-aided design for microfluidic chips based on multilayer soft lithography. In *2009 IEEE International Conference on Computer Design*, pages 2–9. IEEE, IEEE, 2009.

[5] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *ACM Sigmod Record*, volume 19, pages 322–331. ACM, 1990.

[6] Eran Bendavid, Bianca Mulaney, Neeraj Sood, Soleil Shah, Emilia Ling, Rebecca Bromley-Dulfano, Cara Lai, Zoe Weissberg, Rodrigo Saavedra-Walker, James Tedrow, Dona Tversky, Andrew Bogan, Thomas Kupiec, Daniel Eichner, Ribhav Gupta, John Ioannidis, and Jay Bhattacharya. Covid-19 antibody seroprevalence in santa clara county, california. *medRxiv*, 2020.

[7] Greg Book and Neeta Keshary. Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies. Technical report, University of Connecticut, 12 2001.

[8] Sally A. Browning. The Tree Machine: A Highly Concurrent Computing Environment, Computer Science Technical Reports 1980.3760. Technical report, California Institute of Technology, 1980.

[9] J. Burkis. Clock tree synthesis for high performance asics. In *[1991] Proceedings Fourth Annual IEEE International ASIC Conference and Exhibit*, pages P9–8/1, 1991.

[10] Stephen Cameron. *Collision detection by four-dimensional intersection testing.* Oxford University. Computing Laboratory. Programming Research Group, 1990.

[11] Emanuel Carrilho, Andres W Martinez, and George M Whitesides. Understanding wax printing: a simple micropatterning process for paper-based microfluidics. *Anal. Chem.*, 81(16):7091–7095, 2009.

[12] John Anthony Chandler. Assay device for detecting the presence of an analyte involving sequential delivery of reagents through a liquid circuit, November 14 2000. US Patent 6,146,589.

[13] T Chard. Pregnancy tests: a review. *Hum Reprod*, 7(5):701–10, May 1992.

[14] Hong Chen, Jeremy Cogswell, Constantine Anagnostopoulos, and Mohammad Faghri. A fluidic diode, valves, and a sequential-loading circuit fabricated on layered paper. *Lab Chip*, 12(16):2909–2913, 2012.

[15] Jonathan D Cohen, Ming C Lin, Dinesh Manocha, and Madhav Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff, 1995.

[16] Brian Crites, Karen Kong, and Philip Brisk. Diagonal component expansion for flow-layer placement of flow-based microfluidic biochips. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):126, 2017.

[17] Brian Crites, Karen Kong, and Philip Brisk. Directed placement for mvlsi devices. *J. Emerg. Technol. Comput. Syst.*, 16(2), December 2019.

[18] Brian M Cummins, Rukesh Chinthapatla, Frances S Ligler, and Glenn M Walker. Time-dependent model for fluid flow in porous materials with multiple pore sizes. *Analytical Chemistry*, 89(8):4377–4381, 2017.

[19] Nicholas S DeChiara, Daniel J Wilson, and Charles R Mace. An open software platform for the automated design of paper-based microfluidic devices. *Scientific reports*, 7(1):1–8, 2017.

[20] Rachel N Deraney, Charles R Mace, Jason P Rolland, and Jeremy E Schonhorn. Multiplexed, patterned-paper immunoassay for detection of malaria and dengue fever. *Analytical Chemistry*, 88(12):6161–6165, 2016.

[21] Emilio Di Giacomo, Walter Didimo, and Giuseppe Liotta. Radial drawings of graphs: Geometric constraints and trade-offs. *Journal of Discrete Algorithms*, 6(1):109 – 124, 2008. Selected papers from AWOCA 2005.

[22] Wijitar Dungchai, Orawon Chailapakul, and Charles S Henry. A low-cost, simple, and rapid fabrication method for paper-based microfluidics using wax screen-printing. *Analyst*, 136(1):77–82, 2011.

[23] Peter Eades. Drawing Free Trees. Technical report, International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991.

[24] Eclectic Products, Inc. Material safety data sheet. `http://eclecticproducts.com/downloads/sds-e6000-craft-english.pdf`.

[25] Emanuel Elizalde, Raúl Urteaga, and Claudio LA Berli. Precise capillary flow for paper-based viscometry. *Microfluidics and Nanofluidics*, 20(10):135, 2016.

[26] Neil M Ferguson, Derek AT Cummings, Christophe Fraser, James C Cajka, Philip C Cooley, and Donald S Burke. Strategies for mitigating an influenza pandemic. *Nature*, 442(7101):448–452, 2006.

[27] Syrena C Fernandes, Gabriel S Logounov, James B Munro, and Charles R Mace. Comparison of three indirect immunoassay formats on a common paper-based microfluidic device architecture. *Analytical Methods*, 8(26):5204–5211, 2016.

[28] Syrena C Fernandes, Daniel J Wilson, and Charles R Mace. Fabrication of three-dimensional paper-based microfluidic devices for immunoassays. *Journal of visualized experiments: JoVE*, (121), 2017.

[29] Franklin International. Safety data sheet titebond iii ultimate wood glue. `http://www.franklininternational.com/msds/1412.04210m7o0020.pdf`.

[30] Christophe Fraser, Christl A Donnelly, Simon Cauchemez, William P Hanage, Maria D Van Kerkhove, T Déirdre Hollingsworth, Jamie Griffin, Rebecca F Baggaley, Helen E Jenkins, Emily J Lyons, et al. Pandemic potential of a strain of influenza a (h1n1): early findings. *science*, 324(5934):1557–1561, 2009.

[31] PP Galanis, PJW He, IN Katis, AH Iles, Anto JU Kumar, RW Eason, and CL Sones. Local photo-polymer deposition-assisted fabrication of multilayer paper-based devices. *Sensors and Actuators B: Chemical*, 322:128574, 2020.

[32] Xiuqing Gong, Xin Yi, Kang Xiao, Shunbo Li, Rimantas Kodzius, Jianhua Qin, and Weijia Wen. Wax-bonding 3d microfluidic chips. *Lab on a Chip*, 10(19):2622–2627, 2010.

[33] Ariana Gonzalez, Lissette Estala, Michelle Gaines, and Frank A Gomez. Mixed thread/paper-based microfluidic chips as a platform for glucose assays. *Electrophoresis*, 37(12):1685–1690, 2016.

[34] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180. ACM, 1996.

[35] Andreas Grimmer, Qin Wang, Hailong Yao, Tsung-Yi Ho, and Robert Wille. Close-to-optimal placement and routing for continuous-flow microfluidic biochips. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 530–535. IEEE, IEEE, 2017.

[36] Andreas Grimmer, Qin Wang, Hailong Yao, Tsung-Yi Ho, and Robert Wille. Close-to-optimal placement and routing for continuous-flow microfluidic biochips. In *22nd Asia and South Pacific Design Automation Conference, ASP-DAC 2017, Chiba, Japan, January 16-19, 2017*, pages 530–535. IEEE, 2017.

[37] Simon Gruener and Patrick Huber. Imbibition in mesoporous silica: rheological concepts and experiments on water and a liquid crystal. *Journal of Physics: Condensed Matter*, 23(18):184109, 2011.

[38] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[39] Mahiar M Hamedi, Alar Ainla, Firat Güder, Dionysios C Christodouleas, M Teresa Fernández-Abedul, and George M Whitesides. Integrating electronics and microfluidics on paper. *Advanced Materials*, 28(25):5054–5063, 2016.

[40] Henkel. Technical data sheet loctite vinyl, fabric & plastic flexible adhesive. https://dm.henkel-dam.com/is/content/henkel/TDS-1360694-US-Loctite-Vinyl-Fabric-Plastic-Flexible-Adhesive-Carded-Tube-1-fl-oz-2015-07-13.pdf.

[41] David W Hightower. A solution to line-routing problems on the continuous plane. In *Proceedings of the 6th annual Design Automation Conference*, pages 1–24. ACM, 1969.

[42] Seokbin Hong and Wonjung Kim. Dynamics of water imbibition through paper channels with wax boundaries. *Microfluidics and Nanofluidics*, 19(4):845–853, 2015.

[43] Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, pages 82–87. Ieee, 1994.

[44] Yi-Ling Hsieh and Tsung-Yi Ho. Automated physical design of microchip-based capillary electrophoresis systems. In *2011 24th Internatioal Conference on VLSI Design*, pages 165–170. IEEE, 2011.

[45] Philip M Hubbard. Interactive collision detection. In *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*, pages 24–31. IEEE, 1993.

[46] Philip Martyn Hubbard. Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):218–230, 1995.

[47] Boguslaw Jackowski. Graphics-computing the area and winding number for a bézier curve. *TUGboat-TeX Users Group*, 33(1):98, 2012.

[48] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32(4):33, 2013.

[49] Ali Turab Jafry, Hosub Lim, Seong Il Kang, Ji Won Suk, and Jinkee Lee. A comparative study of paper-based microfluidic devices with respect to channel geometry. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 492:190–198, 2016.

[50] Doug L James and Dinesh K Pai. Bd-tree: output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (TOG)*, 23(3):393–398, 2004.

[51] Brent Kalish and Hideaki Tsutsui. Patterned adhesive enables construction of nonplanar three-dimensional paper microfluidic circuits. *Lab on a Chip*, 14(22):4354–4361, 2014.

[52] Mike Kamermans. A Primer on Bézier Curves. https://pomax.github.io/bezierinfo/, 2020. [Online; accessed 1-October-2020].

[53] J Mark Keil. Polygon decomposition. *Handbook of computational geometry*, 2:491–518, 2000.

[54] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[55] James T Klosowski, Martin Held, Joseph SB Mitchell, Henry Sowizral, and Karel Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[56] Hyojin Ko, Jumi Lee, Yongjun Kim, Byeongno Lee, Chan-Hee Jung, Jae-Hak Choi, Oh-Sun Kwon, and Kwanwoo Shin. Active digital microfluidic paper chips with inkjet-printed patterned electrodes. *Advanced Materials*, 26(15):2335–2340, 2014.

[57] Sinan Kockara, Tansel Halic, K Iqbal, Coskun Bayrak, and Richard Rowe. Collision detection: A survey. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 4046–4051. IEEE, 2007.

[58] Edwin H Land. Color printing process, September 8 1964. US Patent 3,147,699.

[59] Charles E. Leiserson. Area-efficient graph layouts (for VLSI). In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 270–281. IEEE Computer Society, 1980.

[60] Gregory G Lewis, Matthew J DiTucci, Matthew S Baker, and Scott T Phillips. High throughput method for prototyping three-dimensional, paper-based microfluidic devices. *Lab on a Chip*, 12(15):2630–2633, 2012.

[61] Ming C Lin and John F Canny. A fast algorithm for incremental distance calculation. In *ICRA*, volume 91, pages 9–12, 1991.

[62] Hong Liu and Richard M Crooks. Three-dimensional paper microfluidic devices assembled using the principles of origami. *J. Am. Chem. Soc.*, 133(44):17564–17566, 2011.

[63] Zhi Liu, Xiaocong He, Jiaxiu Han, Xiuhai Zhang, Fei Li, Ang Li, Zhiguo Qu, and Feng Xu. Liquid wicking behavior in paper-like materials: mathematical models and their emerging biomedical applications. *Microfluidics and Nanofluidics*, 22(11):132, 2018.

[64] Zhi Liu, Jie Hu, Yimeng Zhao, Zhiguo Qu, and Feng Xu. Experimental and numerical studies on liquid wicking into filter papers for paper-based diagnostics. *Applied Thermal Engineering*, 88:280–287, 2015.

[65] Adaris M López-Marzo and Arben Merkoçi. Paper-based sensors and assays: a success of the engineering design and the convergence of knowledge areas. *Lab-on-a-Chip*, 16(17):3150–3176, 2016.

[66] Yao Lu, Weiwei Shi, Lei Jiang, Jianhua Qin, and Bingcheng Lin. Rapid prototyping of paper-based microfluidics with wax for low-cost, portable bioassay. *Electrophoresis*, 30(9):1497–1500, 2009.

[67] Rodrigo G Luque, João LD Comba, and Carla MDS Freitas. Broad-phase collision detection using semi-adjusting bsp-trees. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 179–186. ACM, 2005.

[68] David Mabey, Rosanna W Peeling, Andrew Ustianowski, and Mark D Perkins. Tropical infectious diseases - diagnostics for the developing world. *Nat. Rev. Microbiology*, 2(3):231–240, 2004.

[69] Charles R Mace and Rachel N Deraney. Manufacturing prototypes for paper-based diagnostic devices. *Microfluidics and Nanofluidics*, 16(5):801–809, 2014.

[70] A. Manz, N. Graber, and H.M. Widmer. Miniaturized total chemical analysis systems: A novel concept for chemical sensing. *Sensors and Actuators B: Chemical*, 1(1):244 – 248, 1990.

[71] Andres W Martinez, Scott T Phillips, Manish J Butte, and George M Whitesides. Patterned paper as a platform for inexpensive, low-volume, portable bioassays. *Angewandte Chemie International Edition*, 46(8):1318–1320, 2007.

[72] Andres W Martinez, Scott T Phillips, Zhihong Nie, Chao-Min Cheng, Emanuel Carrilho, Benjamin J Wiley, and George M Whitesides. Programmable diagnostic devices made from paper and tape. *Lab Chip*, 10(19):2499–2504, 2010.

[73] Andres W Martinez, Scott T Phillips, and George M Whitesides. Three-dimensional microfluidic devices fabricated in layered paper and tape. *P. Natl. Acad. Sci. USA*, 105(50):19606–19611, 2008.

[74] Andres W Martinez, Scott T Phillips, George M Whitesides, and Emanuel Carrilho. Diagnostics for the developing world: Microfluidic paper-based analytical devices. *Anal. Chem.*, 82(1):3–10, 2010. PMID: 20000334.

[75] Andres W. Martinez, Scott T. Phillips, George M. Whitesides, and Emanuel Carrilho. Diagnostics for the developing world: Microfluidic paper-based analytical devices. *Analytical Chemistry*, 82(1):3–10, 2010. PMID: 20000334.

[76] Jeffrey McDaniel, Auralila Baez, Brian Crites, Aditya Tammewar, and Philip Brisk. Design and verification tools for continuous fluid flow-based microfluidic devices. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 219–224. IEEE, 2013.

[77] Koichi Mikami. A computer program for optimal routing of printed circuit connectors. *IFIPS Proc., 1968*, 1968.

[78] Christina E Mills, James M Robins, and Marc Lipsitch. Transmissibility of 1918 pandemic influenza. *Nature*, 432(7019):904–906, 2004.

[79] Wajid Hassan Minhass, Paul Pop, Jan Madsen, and Felician Stefan Blaga. Architectural synthesis of flow-based microfluidic large-scale integration biochips. In *Proceedings of the 2012 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pages 181–190, 2012.

[80] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, 1988.

[81] Michael T Osterholm. Preparing for the next pandemic. *New England Journal of Medicine*, 352(18):1839–1842, 2005.

[82] Ian J. Palmer and Richard L. Grimsdale. Collision detection for animation using sphere-trees. In *Computer Graphics Forum*, volume 14, pages 105–116. Wiley Online Library, 1995.

[83] M G Pollack, A D Shenderov, and R B Fair. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab Chip*, 2:96–101, 2002.

[84] Joshua Potter, Philip Brisk, and William H Grover. Using printer ink color to control the behavior of paper microfluidics. *Lab on a Chip*, 19(11):2000–2008, 2019.

[85] Joshua Potter, Philip Brisk, and William H. Grover. Rapid development and optimization of paper microfluidic designs using software automation. *Analytica Chimica Acta*, page in press, 2021.

[86] Joshua Potter, William H Grover, and Philip Brisk. Design automation for paper microfluidics with passive flow substrates. In *Proceedings of the Great Lakes Symposium on VLSI*, pages 215–220, 2017.

[87] Poomrat Rattanarat, Wijitar Dungchai, David Cate, John Volckens, Orawon Chailapakul, and Charles S Henry. Multilayer paper-based device for colorimetric and electrochemical quantification of metals. *Analytical Chemistry*, 86(7):3555–3562, 2014.

[88] Christophe Renault, Jessica Koehne, Antonio J Ricco, and Richard M Crooks. Three-dimensional wax patterning of paper fluidic devices. *Langmuir*, 30(23):7030–7036, 2014.

[89] Amgad R Rezk, Aisha Qi, James R Friend, Wai Ho Li, and Leslie Y Yeo. Uniform mixing in paper-based microfluidic systems using surface acoustic waves. *Lab on a Chip*, 12(4):773–779, 2012.

[90] Jeremy E Schonhorn, Syrena C Fernandes, Anjali Rajaratnam, Rachel N Deraney, Jason P Rolland, and Charles R Mace. A device architecture for three-dimensional, patterned paper immunoassays. *Lab on a Chip*, 14(24):4653–4658, 2014.

[91] Thomas W Sederberg and Tomoyuki Nishita. Curve intersection using bézier clipping. *Computer-Aided Design*, 22(9):538–549, 1990.

[92] Hao Sun, Yuan Jia, Hui Dong, and Longxiang Fan. Graphene oxide nanosheets coupled with paper microfluidics for enhanced on-site airborne trace metal detection. *Microsystems and Nanoengineering*, 5(1):1–12, 2019.

[93] Regina Verena Taudte, Alison Beavis, Linzi Wilson-Wilde, Claude Roux, Philip Doble, and Lucas Blanes. A portable explosive detector based on fluorescence quenching of pyrene deposited on coloured wax-printed μpads. *Lab Chip*, 13(21):4164–4172, 2013.

[94] Jeroen ter Schiphorst, Janire Saez, Dermot Diamond, Fernando Benito-Lopez, and Albertus PHJ Schenning. Light-responsive polymers for microfluidic applications. *Lab on a Chip*, 18(5):699–709, 2018.

[95] S. C. Terry, J. H. Jerman, and J. B. Angell. A gas chromatographic air analyzer fabricated on a silicon wafer. *IEEE Transactions on Electron Devices*, 26(12):1880–1886, 1979.

[96] The Gorilla Glue Company. Safety data sheet - gorilla super glue gel. `https://hk.gorillaglue.com/media/normal/372_Gorilla_Super_Glue_Gel_-_North_America_SDS.pdf`.

[97] The Gorilla Glue Company. Safety data sheet - original gorilla glue. `https://www.gorillatough.com/wp-content/uploads/SDS-Update_Original-Gorilla-Glue-v1.5.pdf`.

[98] Todd Thorsen, Sebastian J Maerkl, and Stephen R Quake. Microfluidic large-scale integration. *Science*, 298(5593):580–584, 2002.

[99] Tsun-Ming Tseng, Mengchu Li, Daniel Nestor Freitas, Travis McAuley, Bing Li, Tsung-Yi Ho, Ismail Emre Araci, and Ulf Schlichtmann. Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 37(8):1588–1601, 2018.

[100] Tsun-Ming Tseng, Mengchu Li, Daniel Nestor Freitas, Amy Mongersun, Ismail Emre Araci, Tsung-Yi Ho, and Ulf Schlichtmann. Columba S: a scalable co-layout design automation tool for microfluidic large-scale integration. In *Proceedings of the 55th Annual Design Automation Conference, DAC 2018, San Francisco, CA, USA, June 24-29, 2018*, pages 163:1–163:6. ACM, 2018.

[101] Tsun-Ming Tseng, Mengchu Li, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. Columba: co-layout synthesis for continuous-flow microfluidic biochips. In *Proceedings of the 53rd Annual Design Automation Conference, DAC 2016, Austin, TX, USA, June 5-9, 2016*, pages 147:1–147:6. ACM, 2016.

[102] Jeffrey D Ullman. *Computational Aspects of VLSI*. W. H. Freeman & Co., USA, 1984.

[103] Marc A Unger, Hou-Pu Chou, Todd Thorsen, Axel Scherer, and Stephen R Quake. Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science*, 288(5463):113–116, 2000.

[104] Qin Wang, Zeyan Li, Haena Cheong, Oh-Sun Kwon, Hailong Yao, Tsung-Yi Ho, Kwanwoo Shin, Bing Li, Ulf Schlichtmann, and Yici Cai. Control-fluidic codesign for paper-based digital microfluidic biochips. In *Proceedings of the 35th International Conference on Computer-Aided Design*, pages 103:1–103:8, 2016.

[105] Sying-Jyan Wang, Katherine Shu-Min Li, and Tsung-Yi Ho. Test and diagnosis of paper-based microfluidic biochips. In *Proceedings of the 4th IEEE VLSI Test Symposium*, pages 1–6, 2016.

[106] Sying-Jyan Wang, Katherine Shu-Min Li, Tsung-Yi Ho, et al. Congestion-and timing-driven droplet routing for pin-constrained paper-based microfluidic biochips. In *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, pages 593–598. IEEE, 2016.

[107] Edward W Washburn. The dynamics of capillary flow. *Physical review*, 17(3):273, 1921.

[108] George M Whitesides, Abraham D Stroock, et al. Flexible methods for microfluidics. *Phys. Today*, 54(6):42–48, 2001.

[109] Daniel J Wilson, Ashok A Kumar, and Charles R Mace. Overreliance on cost reduction as a featured element of sensor design. *ACS Sensors*, 4(5):1120–1125, 2019.

[110] Xerox Corporation. Xerox Safety Data Sheet S-00002. https://safetysheets.business.xerox.com/wp-content/uploads/2017/03/S-00002.en-us.pdf.

[111] Xerox Corporation. Xerox Safety Data Sheet S-9506. https://safetysheets.business.xerox.com/wp-content/uploads/2018/01/S-9506.en-us.pdf.

[112] Xerox Corporation. Xerox Safety Data Sheet S-9511. https://safetysheets.business.xerox.com/wp-content/uploads/2017/03/S-9511.en-us.pdf.

[113] Kailin Yang, Hailong Yao, Tsung-Yi Ho, Kunze Xin, and Yici Cai. AARF: any-angle routing for flow-based microfluidic biochips. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 37(12):3042–3055, 2018.

[114] Ning Yang, Pan Wang, Chun-Yang Xue, Jun Sun, Han-Ping Mao, and Paul Kwabena Oppong. A portable detection method for organophosphorus and carbamates pesticide residues based on multilayer paper chip. *Journal of Food Process Engineering*, 41(8):e12867, 2018.

[115] Chee K Yap. Complete subdivision algorithms, i: Intersection of bezier curves. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 217–226, 2006.

[116] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst. Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, INFOVIS '01, page 43, USA, 2001. IEEE Computer Society.

[117] Gabriel Zachmann. Rapid collision detection by dynamically aligned dop-trees. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 90–97. IEEE, 1998.

[118] Chunsun Zhang, Jinliang Xu, Wenli Ma, and Wenling Zheng. Pcr microfluidic devices for dna amplification. *Biotechnology advances*, 24(3):243–284, 2006.