

# UC Riverside

## UC Riverside Previously Published Works

### Title

Reducing Microfluidic Very Large-Scale Integration (mVLSI) Chip Area by Seam Carving

### Permalink

<https://escholarship.org/uc/item/9vk5x6ms>

### Journal

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(10)

### ISSN

0278-0070

### Authors

Crites, Brian  
Falzone, Cody  
Lopez, Tristan  
et al.

### Publication Date

2021-10-01

### DOI

10.1109/tcad.2020.3033499

Peer reviewed

# Reducing Microfluidic Very Large Scale Integration (mVLSI) Chip Area by Seam Carving

Brian Crites *University of California, Riverside* Cody Falzone *University of California, Riverside*  
 Tristan Lopez *University of California, Riverside* Karen Kong *University of California, Riverside*  
 Philip Brisk *University of California, Riverside*

**Abstract**—Seam carving is an algorithm that analyzes image content and can be used for size reduction in a manner that avoids direct compression or downscaling. Seam carving iteratively identifies horizontal and/or vertical paths of least visual importance and removes them from the image; each path removal reduces the length or width of the image by one row or column of pixels. This paper adapts seam carving to reduce excess area of flow-based microfluidic chips that have been drawn by hand or by computer-aided heuristics without negatively impacting their functionality. The proposed approach leverages domain knowledge, wherein the image to be carved consists of I/O ports, components, and fluid channels, with known and understood fluidic behavior. Three different variants of seam carving are presented: linear, non-linear, and non-rectilinear; experimental results show that non-rectilinear, which is the most general of the three, yields the best results: it improves area utilization by  $8.6x$  and reduces fluid routing channel length by 73% across a set of benchmark microfluidic designs.

**Index Terms**—Microfluidics, Seam Carving, mVLSI

## I. INTRODUCTION

Laboratories-on-a-chip (LoCs) based on continuous fluid flow microfluidics are widely used for a variety of biochemical applications. Through automation and miniaturization, LoCs offer the benefits of higher throughput, lower sample/reagent usage, and reduced likelihood of human error compared to traditional benchtop chemistry methods.

As user requirements and device complexities increase, designers will require CAD tools to cope with increasing device complexity. On the one hand, optimal [1], [2] and near-optimal [3], [4] algorithms have been proposed to generate good quality layouts, they will inevitably suffer from scalability issues unless it is proven that  $P = NP$ ; on the other hand, heuristic methods [5], [6], [7] yield lower-quality layouts with much less execution time overhead.

This paper adapts seam carving [8], an image size reduction technique, to reduce area and channel length of microfluidic layouts produced by fast-running heuristics. The basic premise is to identify seams (paths) through the chip which can be removed without adversely affecting device functionality. This paper presents three variants of the basic seam carving algorithm: linear, non-linear, and non-rectilinear seam carving. The linear and non-linear algorithms require that the layout being carved is rectilinear; while the non-rectilinear variant relaxes this constraint. Linear seam carving restricts seams to be straight-line segments that cross the chip, either from top-to-bottom or left-to-right; non-linear and non-rectilinear seam

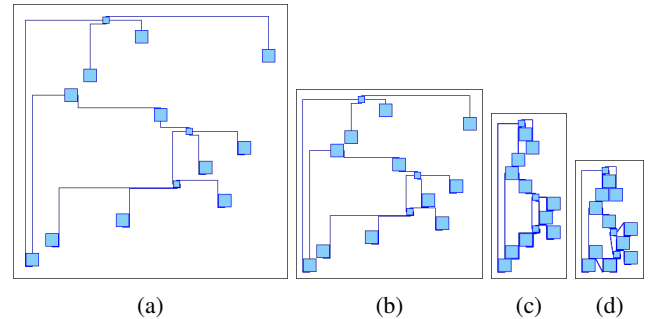


Fig. 1: (a) The AquaFlex-3b benchmark after the baseline placement and routing [5] has completed. The same benchmark using the same placement and routing method is then shown after applying (b) linear seam carving, (c) non-linear seam carving, and (d) non-rectilinear seam carving, as presented in this paper.

carving relax this restriction, to allow seams that feature both rectilinear and diagonal segments.

Fig. 1 presents a motivating example; details of the device's internal components are suppressed. Fig. 1a shows a relatively low-quality rectilinear layout generated by a previously published heuristic [5]. Fig. 1b shows an improved layout, which was derived using linear seam carving; on average, linear seam carving improves area utilization by  $1.4x$  and reduces the average fluid routing channel length by 13%. Fig. 1c shows a better result which was obtained using the more aggressive non-linear seam carving; non-linear seam carving improves area utilization by  $4.28x$  on average, while reducing average fluid routing channel length by 53%. Fig. 1d shows the best result from the application of non-rectilinear seam carving. Non-rectilinear seam carving has a  $8.58x$  improvement in area utilization on average and reduces the average fluid routing channel length by 73%.

This paper is an extension of Ref. [9], which was published in GLSVLSI 2017, and introduced the linear and non-linear seam carving methods. This transactions paper includes the following extensions to the original:

- The introduction of a novel non-rectilinear seam carving method, which overcomes some of the limitations inherent to linear and non-linear seam carving.
- Several new benchmarks from the ParchMint suite [10], [11], which were not available at the time Ref. [9] was published, are included in the experimental evaluation.

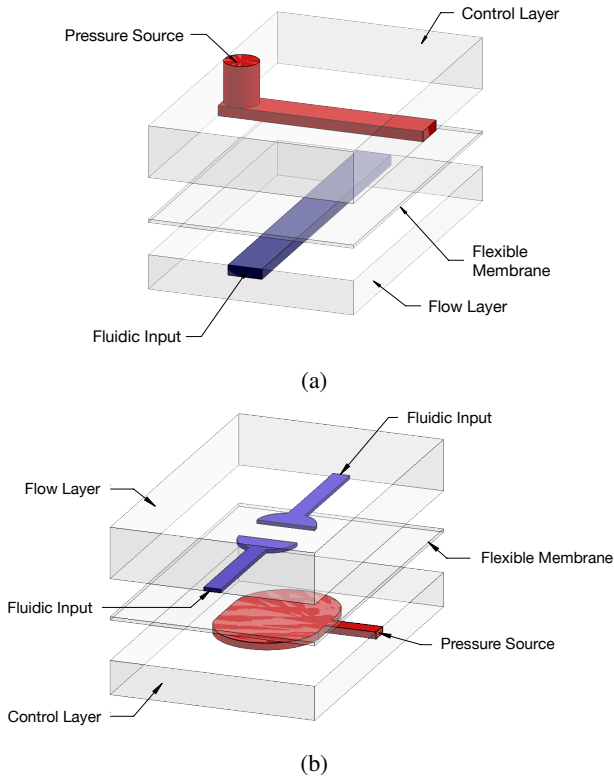


Fig. 2: Cross-sections of elastomeric microvalves [26], [27], fabricated with a flexible membrane placed between two ridged substrates. These microvalves can be designed to be (a) default open where a pressure must be applied to close the fluidic path and (b) default closed where a vacuum must be applied to open the fluidic path.

## II. BACKGROUND AND RELATED WORK

### A. Microfluidic Large Scale Integration

Large scale integration for microfluidics was demonstrated through the development of soft lithography [12], which enabled patterning in flexible polymer materials, such as polydimethylsiloxane (PDMS). This led to the integration of a number of discrete components [13], [14], [15], [16], [17], [18], functional microfluidic structures [19], [20], [21], [22], and, eventually, widespread academic adoption [23], [24], [25].

The use of flexible polymers led to the development of several integrated micromechanical valve technologies which enable controllable manipulation of fluid flow, but require external solenoid valve-based control. Fig. 2a shows the design of a microvalve that can be fabricated using multilayer soft lithography [28]. A number of microfluidic components have been demonstrated using these valves including peristaltic pumps [28], mixers [29], multiplexers [30], and memories [30], [31]. Valves based on this design have achieved very-large scale integration densities [30], [32], [33], [34]; leveraging these technological achievements, a number of fully integrated application-specific [35], [36], [32], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48] and programmable [31], [49] valve-based systems have been demonstrated.

An alternative monolithic membrane microvalve (Fig. 2b) can be realized by sandwiching a thin PDMS membrane between two glass plates with microfabricated channel and valve seat patterns [26]. Monolithic membrane valves have been used in variety of components [27], [50] as well as application-specific [51], [52], [53], [54] and programmable [55], [56] microfluidic systems. While specialized and expensive equipment is needed to create patterned PDMS layers via soft lithography, monolithic membrane valves can be fabricated in virtually any university clean room. Another advantage is that glass has lower native fluorescence than PDMS, making monolithic membrane valve technologies more attractive for applications that require high sensitivity detection.

### B. Seam Carving

A seam is a path of pixels through an image whose removal minimally degrades image quality. Seams can be identified by converting an image into a weighted carving graph, where each vertex represents a pixel and each vertex's weight represents its relative importance to image quality [8]. Seam carving then finds the lowest-cost path (a seam) from one perimeter edge to its opposite and removes the seam from the image. The process repeats until the desired reduction in size is achieved. Seam carving has since been generalized to video, in which 2D seam manifolds are removed from 3D space-time volumes using similar principles [57], [58].

Seam carving removes pixels; it does not adjust colors or apply smoothing to reduce aliasing effects [59]. Seam carving tends to localize aliasing effects to low complexity regions of an image. Seams often loop around “complex” objects within an image, such as text, removing space between the objects before carving through them. Another concern involves images taken of relatively non-complex foreground objects, such as a human face, in a context where the background contains a large number of complex objects: when this occurs, the foreground object tends to be carved at the expense of the background, which inadvertently degrades the viewer's experience [60], [61]. In this work we will show that the above concerns do not exist when the image(s) to be carved are restricted to the domain of mVLSI chip layouts through the careful construction of the carving grid.

### C. mVLSI Placement

Seam carving for mVLSI is presented as a physical design post-processing technique that can reduce device area and channel length. The input to an mVLSI design flow is an architectural netlist specification [62], [63], [10], [11], where vertices represent components (e.g., pumps, mixers) and edges represent channels that transport fluid between components.

In most technologies, fluid flow is only permissible on one substrate layer; thus, it is only possible to lay out planar netlists [5], [64], [1]. It is possible to planarize the netlist by inserting valve-based switches at the intersection points where channels cross [1]. Inserting switches increases both the number of control inputs and the number of control lines to be routed on the control layer.

Given a planar netlist, the next step is to place-and-route the device, which can be decomposed into three distinct problems: flow-layer component placement, flow-layer routing, and control-layer routing; these problems can be solved separately and in sequence [65], [66], [5], [6], [64], [67], [7], or together, which offers the benefit of cross-boundary optimization [1], [2], [4], [68], [69]. This paper describes seam carving as a post-processing technique that can be applied to a physically laid out flow layer, prior to control layer routing. It is also possible to apply seam carving to multi-layer chips by compressing the layers into a single “representative” layout that aggregates all of the layout information across all layers (e.g., both flow and control layers of an active device); seam carving can then be applied to the “representative” layout.

The majority of mVLSI physical design algorithms that have been published to date have been heuristics, which cannot guarantee optimality; in this case, there is at least an opportunity to apply seam carving post-layout to improve the results. The Columba 1.0 [1] and 2.0 [2] layout methods are optimal, based on Integer Linear Programming (ILP) formulations. One could reasonably argue that any post-processing method (seam carving, or otherwise) could not possibly improve a seemingly optimal layout; however, the situation is not quite so straightforward, as we will explain below.

To reduce runtime, the Columba-S (Scalable) ILP [4] imposes some restrictions, which sacrifice global optimality to ensure faster convergence. Similarly, a near-optimal SAT-based mVLSI physical design tool employs search space pruning to achieve tractable results [3]. In both of these cases, seam carving might be able to improve the result.

Seam carving is based on a grid, which creates an inherent tradeoff between algorithmic efficacy and runtime. A finer granularity grid can be expected to yield a more precise carving outcome, but will increase runtime, due to both the time required to search for seams in a denser grid as well as the time required to remove more seams and legalize the outcome of each removal; the memory footprint will also increase, potentially leading to additional memory-related performance issues. Most other mVLSI physical design frameworks are also grid-based, and must deal with similar tradeoffs between solution quality and runtime, vis-a-vis grid density [1], [2], [3], [4]. While gridless routing for semiconductor VLSI is a mature topic [70], [71], [72], [73], a more comprehensive adaptation of gridless layout for mVLSI chips is beyond the scope of this paper and is left open for future work.

### III. PRELIMINARIES

The input to seam carving is a layout (i.e., a placed and routed mVLSI architecture)  $A = (m, n, C, R, \Delta)$ , where  $C$  is a set of placed components,  $R$  is a set of routed channels,  $n$  and  $m$  are the respective height and width of the layout, and  $\Delta \geq 0$  is an optional parameter that adds white space around each component to improve routability.

Each microfluidic component  $c_i \in C$  is represented as a bounding box  $c_i = (x_i, y_i, w_i, h_i)$ : point  $(x_i, y_i)$  is the upper-left corner of the component, and  $h_i + 2\Delta$  and  $w_i + 2\Delta$  are its respective height and width.

TABLE I: Symbols and notation used throughout the paper.

Variable	Description
$n$	Height of the device
$m$	Width of the device
$A$	Set of components, routes, and device information
$C$	Set of placed components
$R$	Set of routed channels
$\Delta$	(Optional) component buffer space
$c_i$	$i$ 'th component of $C$
$(x_i, y_i)$	$i$ 'th component's upper left corner
$h_i$	$i$ 'th component's height
$w_i$	$i$ 'th component's width
$r_i$	$i$ 'th routing channel in $R$
$r_{i,j}$	$j$ 'th channel segment of the $i$ 'th channel of $R$
$p_u, p_v$	Channel segment endpoints
$B_x$	Boolean array of possible vertical seams
$B_y$	Boolean array of possible horizontal seams
$G$	$m \times n$ Boolean grid of possible seam candidates
$S$	Set of seams
$s_i$	Seam
$s_{i,j}$	Seam segment
$q_u$	Seam segment endpoint

Each fluid channel  $r_i \in R$  is defined as a set of straight-line channel segments that are physically connected to form a tree. The  $j$ 'th channel segment  $r_{i,j}$  of channel  $r_i$  is defined as a pair of points  $r_{i,j} = (p_u, p_v)$ , where  $p_u = (x_u, y_u)$  and  $p_v = (x_v, y_v)$ . No channel segment may intersect a component unless it is explicitly connected to that component through a *port* on that component's perimeter.

Multi-segment channels must intersect. We require that any intersection point between two or more channels be an explicit endpoint of all channels involved. For example, consider a T-junction consisting of two segments:  $r_{1,1} = ((1, 1), (10, 1))$  and  $r_{1,2} = ((5, 1), (5, 5))$ . The intersection occurs at point  $(5, 1)$ , which is on channel segment  $r_{1,1}$ , but is not an endpoint; this is not allowable.  $r_{1,1}$  would need to be split into two distinct segments:  $((1, 1), (5, 1))$  and  $((5, 1), (10, 1))$ .

A layout is *rectilinear* if all channel segments are horizontal or vertical; otherwise (i.e., if there is at least one diagonal channel segment), the layout is non-rectilinear. The linear and non-linear seam carving algorithms described in this paper are limited to rectilinear layouts; non-rectilinear seam carving algorithm can handle non-rectilinear layouts as well.

A layout is *legal* (or *valid*) if it is planar: no components may overlap, two distinct channels may not intersect, and no channel segment may intersect the area allocated to a component. The one exception is when the channel segment ends at an input or output port of a component; then, it is allowed to intersect the component on its perimeter at the location of the port. Layouts that do not satisfy these properties are illegal and are not admissible for carving.

In microfluidic devices all physical space can be classified into three categories: components, channels, and unused space. Components have a fixed height and width; non-rectangular components are represented by the smallest rectangular bounding box that enclose them. Fluid channels can have any length, number of segments, or fanouts, as long as they provide a continuous flow of fluid between all of their incident components. Channel and channel segment length can be reduced



without altering chip functionality. Unused space is otherwise superfluous and can be eliminated with one exception: unused space defined by the parameter  $\Delta$  is implicitly treated as being internal to each component.

A *seam* is a path through the architecture that connects one perimeter edge to its opposite and contains only points that are valid for removal. This ensures that correct device functionality is maintained once a seam is removed. Seams are not allowed to carve through components, as we assume that doing so would adversely affect chip functionality. Seams may always carve through unused space, which does not perform any functionality. Seams may carve through channels (i.e., to reduce their length), but may not carve through channel intersection points: removing an intersection point would require a subsequent post-processing step to re-insert it and re-route its incident fluid channel segments; we prefer to avoid this overhead.

Let  $S$  be a set of seams. Each seam  $s_i \in S$  is an ordered sequence of  $k$  straight-line *seam segments* sharing common endpoints. Formally,  $s_i = \{q_1, q_2, \dots, q_k, q_{k+1}\}$ , where  $s_{i,j} = (q_i, q_{i+1}), 1 \leq i \leq k$  is the  $j$ 'th segment from point  $q_j$  to  $q_{j+1}$ . A seam from the left to the right perimeter satisfies  $x_1 = 0, x_{k+1} = m$ ; a seam from the bottom to the top perimeter satisfies  $y_1 = 0, y_{k+1} = n$ . A *Linear Seam* consists of one horizontal or vertical segment; a *Nonlinear Seam* consists of an ordered sequence of alternating horizontal and vertical seam segments obtained from a rectilinear layout; and a *Non-rectilinear seam* has the same properties as a Nonlinear seam without the rectilinear requirement.

The above notation is collected in Table I for ease of reference. The following sections present algorithms for linear, non-linear, and non-rectilinear seam identification and carving.

Seam carving is directional: *horizontal seam carving* proceeds along the  $x$ -axis, while *vertical seam carving* proceeds along the  $y$ -axis. The initial segment of each seam that is generated will be *perpendicular* to the carving direction: without loss of generality, if we are carving in the horizontal direction, the first seam segment will be vertical, i.e., it will have the form  $((x_i, 0), (x_i, j))$  for some  $j > 0$ ; the last segment of the seam will also be vertical. In the degenerate case, all linear seams are perpendicular to the carving direction.

#### IV. LINEAR SEAM CARVING

*Linear seam carving* restricts seams to be single segments that span the chip in the horizontal and vertical directions. Fig. 3a shows an example mVLSI chip with a loose placement and ample white space. Fig. 3b shows four horizontal seams, two of which intersect fluid channels in the center of the chip. Fig. 3c shows the smaller chip after the four seams are removed. Device functionality is not altered, and the channel connecting the two components is shortened but not disrupted.

##### A. Seam Identification

Linear seam carving employs two Boolean arrays,  $B_x$  and  $B_y$ , which represent removable vertical and horizontal seams. Without loss of generality, as we move along the  $x$ -axis,  $B_x[i]$  represents a vertical line containing all points within

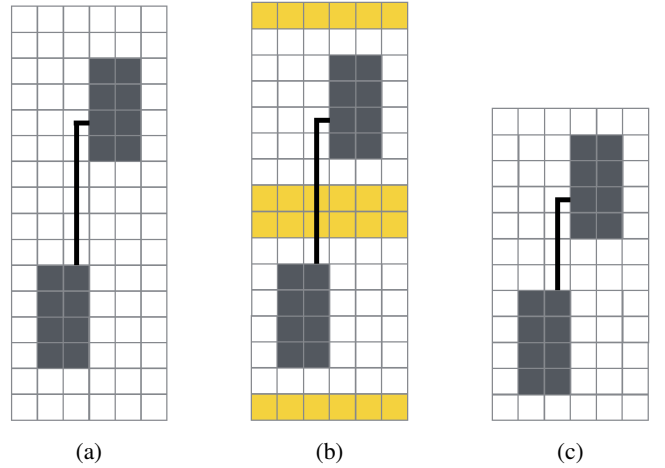


Fig. 3: (a) A laid out mVLSI chip; (b) seam identification ( $\Delta = 1$ ); (c) the chip after seam removal.

the component having  $i$  as the  $x$ -coordinate. Both arrays are initialized to  $B_x[1 : m] = B_y[1 : n] = True$ .

The algorithm identifies vertical and horizontal seams for removal separately. Any index  $i$  for which  $B_x[i] = True$  represents a vertical seam that could be removed. Horizontal seams are identified similarly, using  $B_y$  and the  $y$ -coordinates of components and channel segments. To identify vertical seams, the algorithm iterates through all components  $c_i \in C$  setting  $B_x[x_i - \Delta : x_i + w_i + \Delta] = False$ ; this disallows any seam that cuts through a component. For each channel  $r_i \in R$ , and for each channel segment  $r_{i,j} = (p_u, p_v) \in r_i$  the algorithm sets  $B_x[x_u] = False$  and  $B_x[x_v] = False$  to prevent removal of the entire channel segment.

Without loss of generality, a vertical seam could cut through a horizontal channel segment, shortening it by one unit; however, a vertical seam that cuts through a vertical channel segment would cut through the entire segment. If a channel segment of a pre-specified length is required (e.g., to achieve a chemical separation), then a portion of that channel can be treated as a component with  $\Delta = 0$ .

##### B. Seam Carving

Each index  $j \in \{0, \dots, m\}$  where  $B_x[j]$  is *True* is a removable vertical seam. Each component  $c_i \in C$  such that  $x_i > j$  is shifted left to fill the space removed by the seam; the height and width of  $c_i$  remain unchanged. Channel segments completely to the right of the removed seam are shifted left by one grid point. The right endpoint is shifted left by one grid point. The final step is to reduce the list of possible seam candidates  $B_x$  by one by setting  $B_x[k] = B_x[k + 1], j \leq k \leq m$ , and decrementing  $m$ . This process then repeats similarly for all horizontal seams,  $0 \leq j \leq n$  where  $B_y[j]$  is *True*.

#### V. NON-LINEAR SEAM CARVING

Non-linear seam carving eliminates the restriction that seams must be exclusively horizontal or vertical segments. Seams are still required to begin at one perimeter edge and end at the opposite edge. This increases opportunities for seam

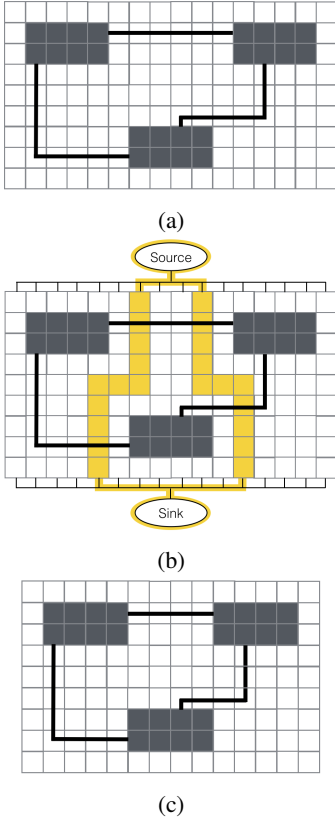


Fig. 4: (a) A placed and routed mVLSI chip; (b) identification of nonlinear seams from the *Source* to *Sink* that can be removed without impact on chip functionality; this example sets  $\Delta = 1$ ; (c) the identified nonlinear seams are removed

removal and can lead to substantially smaller chip designs. Fig. 4a shows an example mVLSI chip with a loose placement and ample white space. Fig. 4b shows two non-linear seams, whose removal yields a smaller chip depicted in Fig. 4c.

#### A. Seam Identification

Seam identification employs an  $m \times n$  Boolean grid  $G$  to determine if a given point is a candidate for seam carving. Initially  $G[1 : m][1 : n] = True$ . For each component  $c_i \in C$  at position  $(x_i, y_i)$  the algorithm sets  $G[x_i - \Delta : x_i + w_i + \Delta][y_i - \Delta : y_i + h_i + \Delta] = False$ , rendering these points invalid for inclusion in a seam. For each channel  $r_i \in R$ , and for each channel segment  $r_{i,j} = (p_u, p_v) \in r_i$  the algorithm sets  $G[x_u][y_u] = G[x_v][y_v] = False$  to prevent removal of the entire channel segment and prevent the need to re-route a segment.

*Non-linear seam carving* retains the directional approach of its linear counterpart, but relaxes the requirement that each seam consists of one horizontal or vertical segment that spans the entire chip. Seams are first identified along the  $x$ -axis, with an artificial *source* connected to all grid positions  $G[j][1], 1 \leq j \leq m$  and an artificial *sink* connected to all grid positions  $G[j][n], 1 \leq j \leq m$ ; the source and sink nodes are omitted in Fig. 4 to conserve space. A Maze Router, such as Lee's Algorithm [74], is repeatedly called to identify valid seams

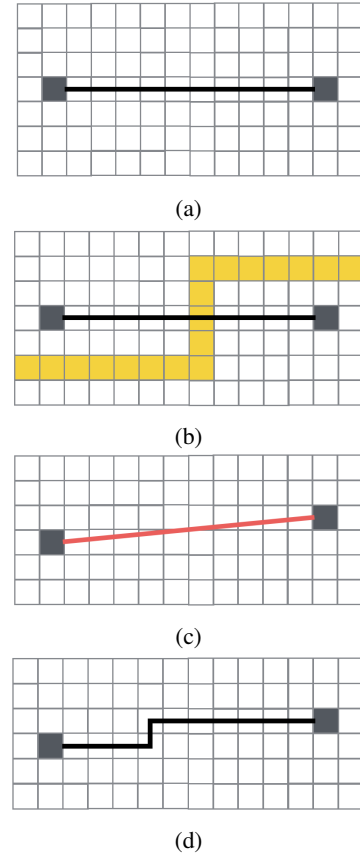


Fig. 5: (a) Two placed and routed components connected by a horizontal channel comprising one segment; (b) a rectilinear seam is identified that crosses the channel; the seam travels horizontally below the component on the left-side and above the component on the right-side of the chip; (c) removing the seam shifts the component on the right up by one grid position, but does not shift the component on the left; the channel segment that connects the two components is no longer rectilinear; (d) the correction is to re-route the channel as a post-processing step, however we explicitly disallow this option as routability cannot be guaranteed in the general case.

through grid entries that are marked *True*; one valid seam at a time is removed. This process iterates until no further valid seams can be found, and then repeats along the  $y$ -axis.

#### B. Perpendicular Carving

Non-linear seam carving requires special handling of channel segments that run perpendicular to the carving direction; we refer to this general situation as *perpendicular carving*.

Fig. 5a shows an example that carves along the vertical axis through a horizontal channel segment  $r_{i,j} = (p_u, p_v)$ , such that  $y_u = y_v$  and  $x_u < x_v$ . Nonlinear seam  $s_i$ , identified in Fig. 5b, consists of three seam segments: a horizontal seam segment  $s_{1,1}$  below the horizontal channel segment; a vertical seam segment  $s_{1,2}$  that cuts through the horizontal channel segment at vertical position  $x_w$ ; and a horizontal seam segment  $s_{1,3}$  above the horizontal channel segment.

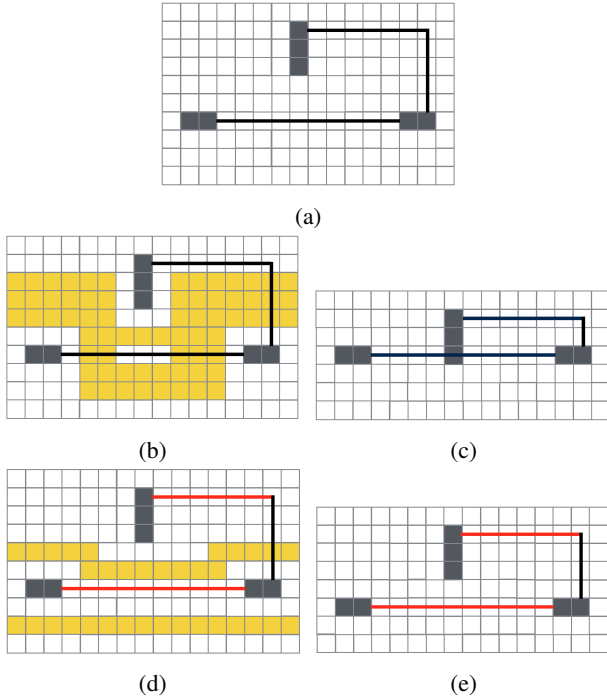


Fig. 6: (a) A placed and routed mVLSI chip; (b) when carving along the  $y$ -axis ( $\Delta = 1$ ), a set of non-linear seams are found that cross a perpendicular (horizontal) segment; (c) removal of the preceding seams with these perpendicularly carved segments yields an invalid layout; (d) to prevent this, the cells containing perpendicular segments are marked as unusable by the seam (all perpendicular segments  $r_{i,j} = (p_u, p_v)$  shown in red); (e) removal of non-linear seams that do not cross the perpendicular segment and therefore cannot introduce perpendicular carve segments yields a legal layout.

Removal of the seam shifts the position of the left endpoint  $p_v$  to a new point, denoted  $p'_v$ , such that  $x'_v = x_v - 1$  and  $y'_v = y_v$ ; we let  $r'_{i,j} = (p_u, p'_v)$  denote the adjusted channel segment. In Fig. 5c,  $r'_{i,j}$  is a diagonal seam, which is not compatible with non-linear seam carving's requirement that all channel segments be rectilinear. In Fig. 5d  $r'_{i,j}$  is instead replaced with three rectilinear segments:  $r_{i,j}^{(1)} = (p'_v, (x'_v, y_w))$ ,  $r_{i,j}^{(2)} = ((x'_v, y_w), (x_v, y_w))$ , and  $r_{i,j}^{(3)} = ((x_v, y_w), p_u)$ .

Perpendicular carving can create collisions between a component and a perpendicular channel. Fig. 6a shows a placed and routed mVLSI chip. In Fig. 6b vertical carving yields three (partially overlapping) seams, two of which cross a horizontal (perpendicular) channel segment; the third seam occupies one of two grid elements of white space that separate the horizontal seam from a  $1 \times 3$  rectangular component above it; all three seams carve through a vertical segment to the right of the  $1 \times 3$  component.

If  $\Delta = 1$ , the rectangular component can be shifted down by one grid position; shifting it further will cause it to intersect the horizontal channel segment. In Fig. 6c, all three seams have been removed. On the right-hand-side of the image, the height of the vertical seam segment has been reduced by 3 units, which causes the  $1 \times 3$  component to shifted down by 3 units

as well; however, only one of the three seams carved through the white space separating the now-shifted component and the horizontal (perpendicular seam segment); consequently, the shifted component intersects the horizontal segment, causing a layout violation. While this example represents one specific case, it readily generalizes to encompass any seam whose orientation is perpendicular to the carving direction.

This problem can be rectified by preventing non-linear seams from carving through channel segments that are perpendicular to the carving direction. Applying nonlinear seam carving with this restriction to the example in Fig. 6a yields a smaller set of seams, shown in Fig. 6d, and whose removal depicts a legal layout shown in Fig. 6e. This restriction ensures that the number of units that can be removed from the vertical seam on the right cannot exceed the number of units that the component in question can be safely shifted down in the vertical direction.

Without loss of generality, assume that we are carving in the  $y$ -direction and let  $r_{i,j}$  be a horizontal channel segment, which implies that  $y_u = y_v$ . To implement the aforementioned restriction, the algorithm sets  $G[x_u : x_v][y_u] = False$  prior to carving. When carving in the  $x$ -direction, carving through vertical channel segments can be suppressed similarly.

### C. Seam Carving

Components or channel segments that lie to the right (when carving along the  $x$ -axis) or below (when carving along the  $y$ -axis) the seam are moved left or up one unit respectively.

Recall that that we define the origin to be the upper-left corner of the grid. Without loss of generality, assume that we are carving along the  $x$ -axis and we identify a seam that includes a vertical segment  $s_i = (q_u, q_v)$ , where  $x_u = x_v$  and  $y_u < y_v$ . For example, the left shift applies to a component  $c_i \in C$  which exists to the right of the seam (i.e.,  $x_i > x_u$ ) and whose top edge lies between the segment endpoints ( $y_u \leq y_i \leq y_v$ ). This is shown, for example, in Fig. 4b and Fig. 4c: the component in the upper-left is to the left of both seams and is not shifted; the component at the bottom is to the right of one seam and to the left of the other, and is shifted left by one grid point; the component in the upper-right is to the right of both seams and is shifted left by two grid points. Channel and channel segment movements are governed by analogous rules. This process then repeats similarly along the  $y$ -axis.

## VI. NON-RECTILINEAR SEAM CARVING

Non-rectilinear seam carving is similar to non-linear seam carving in that seams are required to run from one perimeter edge to the opposite edge and that they may contain multiple alternating horizontal and vertical segments. Non-rectilinear seam carving relaxes two requirements that nonlinear seam carving requires: it eliminates (1) the need to invalidate routes that run perpendicular to the carving direction, and (2) the requirement that channel segments themselves be rectilinear. Relaxing these requirements increase the number of seams that can be removed. Care must be taken to prevent the creation of routes that cross component boundaries; non-rectilinear seam carving includes a technique to identify and undo the removal of seams that result in design rule violations.

NRSC Main Function:

**Require:**  $(C, R, n, m)$ : a placed and routed mVLSI netlist  
 1:  $G \leftarrow InitializeGridGraph(C, R, n, m)$   
 2:  $NRSC\_CarveHorizontal(G, C, R)$   
 3:  $NRSC\_CarveVertical(G, C, R)$

NRSC\_CarveHorizontal Function:

**Require:**  $(G, C, R)$ : Grid graph, Components, and Channels  
 1:  $src \leftarrow AddSourceAtTop(G)$   
 2:  $sink \leftarrow AddSinkAtBottom(G)$   
 3:  $S \leftarrow MazeRoute(G, src, sink)$   
 4: **while**  $|S| > 0$  **do**  
 5:   **for all**  $c_i \in C$  **do**  
 6:     **if**  $RightOfSeam(c_i, S)$  **then**  
 7:       #  $x_i$ : x-coordinate of the upper-left corner of  
       component  $c_i$  (Table I)  
 8:        $x_i \leftarrow x_i - 1$   
 9:     **end if**  
 10:   **end for**  
 11:   **for all**  $r_i \in R$  **do**  
 12:     **for all**  $r_{i,j} \in r_i$  **do**  
 13:       #  $p_u = (x_u, y_u)$  and  $p_v = (x_v, y_v)$ : endpoints of  
       channel segment  $r_{i,j}$  (Table I)  
 14:       **if**  $RightOfSeam(x_u, S)$  **then**

15:         $x_u \leftarrow x_u - 1$   
 16:        **end if**  
 17:        **if**  $RightOfSeam(x_v, S)$  **then**  
 18:          $x_v \leftarrow x_v - 1$   
 19:        **end if**  
 20:     **end for**  
 21:   **end for**  
 22:   **if**  $InvalidLayout(C, R, S)$  **then**  
 23:     **for all**  $c_i \in C$  **do**  
 24:       #  $x_i$ : x-coordinate of the upper-left corner of  
       component  $c_i$  (Table I)  
 25:        $x_i \leftarrow x_i + 1$   
 26:     **end for**  
 27:     **for all**  $r_i \in R$  **do**  
 28:       **for all**  $r_{i,j} \in r_i$  **do**  
 29:         #  $p_u = (x_u, y_u)$  and  $p_v = (x_v, y_v)$ : endpoints  
        of channel segment  $r_{i,j}$  (Table I)  
 30:          $x_u \leftarrow x_u - 1$   
 31:          $x_v \leftarrow x_v - 1$   
 32:       **end for**  
 33:     **end for**  
 34:   **end if**  
 35:    $S \leftarrow MazeRoute(G, src, sink)$   
 36: **end while**  
 37:  $G \leftarrow RemoveSourceAndSink(G, src, sink)$

Fig. 7: Pseudocode for the main function and horizontal carving function of NRSC. Vertical carving is similar to horizontal carving, with several exceptions: source and sink nodes are inserted on the left and right sides of the grid graph; horizontal carving checks if components and channel segments are below the seam being carved; if so, their y-axis values are decremented.

### A. Seam Identification

Non-rectilinear seam carving retains the directional approach of its linear and non-linear counterparts; the seams remain rectilinear, although channel segments may become diagonal. Non-rectilinear seam carving employs the same  $m \times n$  Boolean grid  $G$ , which is invalidated at all points within a component and at the endpoints of each channel segment. Seams are first identified along the  $x$ -axis, using an artificial *source* and *sink* connected to their respective perimeter grid positions, Lee's Algorithm is then repeatedly called to identify seams from *source* to *sink*, until no valid paths remain. This process then repeats along the  $y$ -axis.

Fig. 8a and 8b depict the process of seam identification and removal in the presence of a non-rectilinear channel segment. The removal of the seams yield a smaller layout, shown in Fig. 8c.

### B. Seam Carving

Non-rectilinear seam carving is similar to nonlinear seam carving, with the additional requirement that design rule violations must be identified and rectified. Since non-rectilinear seam carving does not invalidate channel segments that run perpendicular to the carving direction, a seam's removal may shift one endpoint of the channel, but not the other. As shown in Fig. 8c to 8e this could cause a channel to intersect a

component, creating a design rule violation that could not occur under linear or nonlinear seam carving.

When a seam is removed, each channel segment that is shifted must be checked against all other channel segments and components to determine if overlap occurs; component overlap includes the buffer space  $\Delta$ . If no intersections are found, then the seam removal is valid and the process continues; however, if removing the seam induces a design violation, then all components and channel segments that were shifted are restored to their pre-seam-removal positions, as illustrated in Fig. 8f. The seam whose removal caused the design violation is invalidated on the routing grid to ensure that it is not identified again subsequently. Since the design validity check is performed after each seam is removed, it is always possible to restore the state of the device to a valid layout. In the extreme case, all seam removal attempts fail, and the initial layout of the device is returned unmodified.

### C. Discussion

All three seam carving methods identify paths (seams) from one side of the grid to the other. LSC, being the most restrictive of the three methods, uses a Boolean Matrix to represent free space, whereas NLSC and NRSC explicitly construct a graph and employ a maze router to identify seams. Any linear seam that LSC identifies would exist as a source-to-sink path in the graph constructed by either NLSC or NRSC. There is no

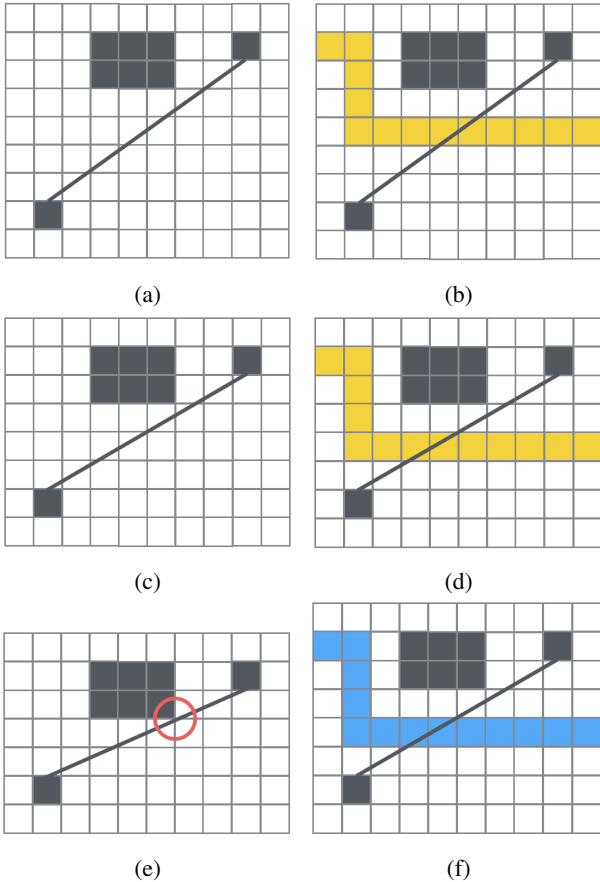


Fig. 8: (a) A placed and routed mVLSI chip; (b) when carving along the  $y$ -axis ( $\Delta = 1$ ), a seam is found that crosses a channel segment; (c) the removal of that seams yields a device with no violations; (d) a second seam is identified that crosses the same channel segment; (e) removal of this seams yields an invalid layout, as the channel segment now intersects a placed component; (f) to rectify the error, the seam is re-introduced into the system, the components and channel segments are shifted back, and the seam is invalidated to prevent subsequent attempts to remove it.

guarantee that the maze router would return this exact path; the solution is guaranteed to be optimal, given the restrictions imposed on the graph (e.g., the restrictions that a non-linear seam cannot cross a channel segment that is perpendicular to the carving direction). Along similar lines, any rectilinear seam that NLSC could identify would exist as a source-to-sink path in the graph constructed by NRSC; once again, there is no guarantee that the maze router would return that path.

There is one minor, but important, caveat involving NRSC. While repeated calls to the maze router aims to identify the maximum number of carvable seams in each direction; however, we have also seen that some non-rectilinear seams cannot be carved, as they result in illegal layouts. We presently lack a method to encode this type of information into the maze router, and thus we cannot guarantee that repeated calls will return the maximum number of legally carvable seams. To partially address this situation, we first run NLSC prior

to NRSC, as all rectilinear seams identified by NLSC are guaranteed to be legally carvable. We then run NRSC to identify non-rectilinear seams, and we carve as many of them as we legally can. It is not immediately clear if the number of carvable non-rectilinear seams depends on the order in which the seams are carved, but it seems quite likely to be the case.

## VII. EXPERIMENTAL RESULTS

We implemented the linear (LSC), non-linear (NLSC), and non-rectilinear (NRSC) seam carving methods in C++ within a larger, internally-developed microfluidic design toolchain; The NRSC method utilizes the Lemon graph library to construct a Boolean grid and identify seams [75]. All experiments were run on an Amazon Web Services' *r5.2xlarge* instance (8 vCPU Xeon Platinum 8175 (Skylake), 64 GB RAM) with the primary constraint being memory.

We used a subset of the ParchMint benchmark suite [10], [11] that is compatible with our toolchain:

- **HIV1**: A bead-based HIV1 immunoassay [76];
- **GPMFD**: A general-purpose software-programmable microfluidic device comprising a mixer, fluidic memory, I/O, and washing pathway [31];
- **MGG**: a molecular gradient generator [77];
- **Chromatin**: a device that performs chromatin immunoprecipitation [78];
- **AquaFlex-3b** and **-5a**: proprietary netlists provided by Microfluidic Innovations LLC [79], a company that is no longer in operation; and
- **Synthetic 1-7**: netlists derived from sequencing graph assays models originally intended to evaluate electrowetting microfluidic chips [80].

To the best of our knowledge, no individual microfluidic physical design flow has been able to successfully place-and-route all of the ParchMint benchmarks. This subset of Parchmint benchmarks, evaluated in Fig. 9, includes the full set of microfluidic netlists used for experimental evaluation in the original seam carving publication [9]. We used a previously-published deterministic mVLSI planar placement and routing method [5] to generate the initial layout, prior to seam carving.

While the benchmarks include those originally published in Ref. [9], there are two experimental differences of note. First, all experimental results presented here use  $\Delta = 0$ , as opposed to the  $\Delta = 5$  value used in the original publication. Second, minor modifications were made to the benchmarks when they were introduced into the ParchMint suite. Specifically the scales used in the devices were modified to bring them in line with the other benchmarks. As all the results were recalculated based on the updated benchmark set and with the new parameters, we expect this to have no measurable impact on the results and see no major differences in the results.

We evaluate the impact of the different seam carving methods in terms of area utilization and average channel length. Utilization is the percentage of the total device area occupied by microfluidic components and fluid channels; average channel length qualifies the space allocated to routed fluid channels that each seam carving method was able to remove. We also report the number of seams carved by each algorithm, noting



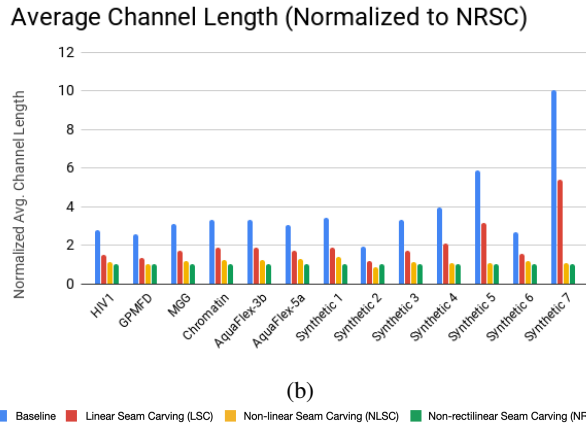
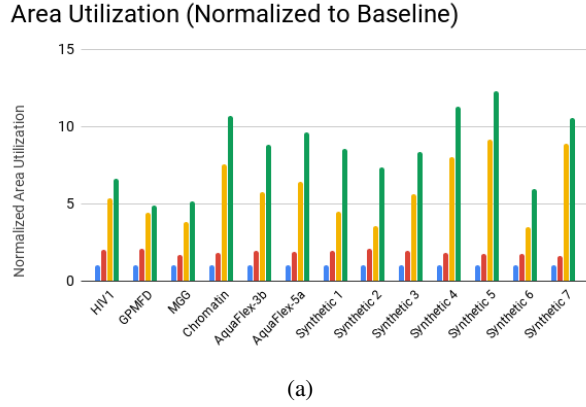


Fig. 9: Experimental comparison of the three seam carving methods to the planar placement and routing baseline [5]. Metrics reported are (a) area utilization normalized to the baseline and (b) average channel length normalized to NRSC.

that LSC and NLSC carve all seams that they identify, while NRSC may identify more seams than it can carve.

To avoid technology-specific bias, we impose a uniform grid structure on the layout. We compute device area and channel length in terms of the number of occupied units (cells). We report the runtime of each of the three seam carving methods.

In terms of area utilization and the number of seams carved NRSC dominates NLSC, which in turn dominates LSC; in terms of average channel length, NLSC and NRSC dominate LSC, with NRSC achieved a shorter average channel length than NLSC for all but one benchmark (Synthetic 2). In short, reducing the number of grid points marked as invalid for seam identification increases the number of seams available for carving. The runtime of NLSC exceeds that of LSC due to its use of a maze router to identify viable seams, while the runtime of NRSC exceeds that of NLSC due to repeatedly re-validate the layout after each seam removal attempt.

#### A. Utilization

Fig. 9a reports the normalized utilization for each benchmark after planar placement and routing [5], and then followed by LSC, NLSC, and NRSC as post-processing steps. The baseline device layout produced by planar placement and routing has an average utilization of 2.9%. Applying LSC

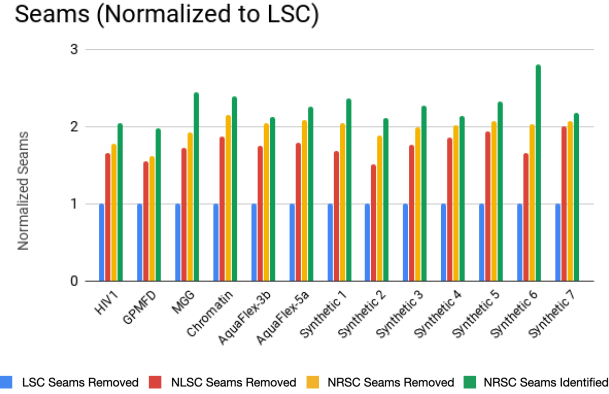


Fig. 10: The number of seams identified and carved by LSC, NLSC, and NRSC; results are normalized to LSC. LSC and NLSC carve all seams that they identify; NRSC identifies more seams than it can carve due to layout violations.

yielded an average utilization improvement of 1.83x to 5.3%. NLSC provides a larger increase utilization as its able to find seams that cross channels and follow non-linear paths between components, increasing the number of possible seams. NLSC increases utilization by 5.6x to 16.2% when compared to the baseline. NRSC, the most effective method, increases the average utilization by 8.1x to 23.5%. As noted previously, NRSC accrues these benefits for two key reasons. The first reason is that removal of rectilinear constraints, creates many opportunities to identify seams. The second is that NRSC invalidates fewer points on the grid than LSC and NLSC, which increases the number of seams available for carving.

#### B. Channel Length

Fig. 9b reports the normalized average fluid channel length per benchmark. The baseline device layout produced by planar placement and routing [5] has an average channel length of 285 units. LSC reduces this average by 33.8% to 189 units when compared to the baseline; NLSC reduces the average by 68.2% to 91 units; and NRSC reduces it by 72.9% to 77 units. The explanation for NRSC's superior results is identical to what was described in the preceding subsection.

For Synthetic 2, NLSC has a smaller average route length, but a larger area, than NRSC. In NRSC its possible for seams to be identified which cut through channel segments such that those which were originally straight-line deflects to an angle, technically lengthening them; this phenomenon occurs, for example, in Fig. 5. In the case of Synthetic 2, this occurred so many times that the average seam length increased, compared to NLSC which ensures that all channel segments remain either horizontal or vertical after carving.

#### C. Seam Identification & Removal

Fig. 10 shows the number of seams identified by LSC (Blue), NLSC (Red) and NRSC (Yellow and Green); results are normalized to LSC. For NRSC, we differentiate between the number of seams identified by the maze router (Green)



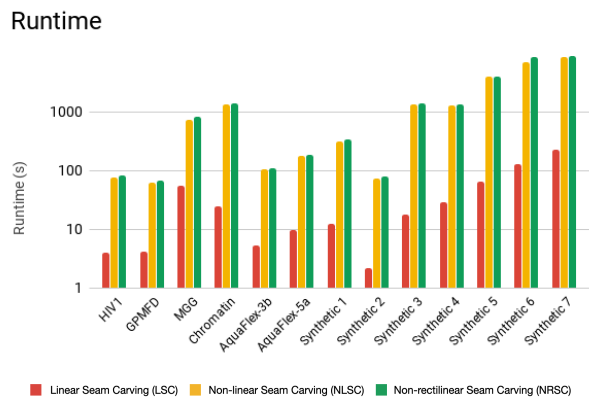


Fig. 11: Runtimes of the three seam carving methods.

and the number of seams that were carved without inducing a layout violation (Yellow). On average, NLSC carves 2.48x more seams than LSC, NRSC successfully carves 2.83x more seams than LSC; NRSC carves 1.14x more seams than NSRC; and NRSC identifies 1.15x more seams than it can carve.

Synthetic 6 is an interesting case for NRSC: the ratio of seams identified to seams carved is much higher in comparison to the other benchmarks. The initial layout places the majority of components along the diagonal of the chip area, with a rather dense network of routing channels. Thus, utilization along the diagonal region is high, while being considerably lower at the peripheral regions of the chip. The excess unused space in the peripheral region allows for the identification of a large number of seams; however, as these seams also cut through the diagonal, attempts to carve them yield layout violations. This observation correlates with the low utilization for Synthetic 6, relative to the other benchmarks (including Synthetic 7, which is similarly sized), as shown in Fig. 9a.

#### D. Runtime

The average runtime per benchmark for the planar placer and router was 718.29 seconds. The average runtime per benchmark for LSC was 42.37 seconds, increasing the average runtime by 5.90% as shown in Fig. 11.

Both NLSC and NSRC run significantly slower than both planar placement and routing and LSC: the average runtime of NLSC was 1959.49 seconds (2.73x slower than planar placement and routing) while the average runtime of NRSC was 2096.79 seconds (2.92x slower than planar placement and routing). The percentage overhead seems to be inversely proportional to device size: for the smallest device (Synthetic 2) the runtimes of LSC, NLSC, and NSRC were 2.07x, 7.60x, and 8.29x, respectively, greater than the time required to compute the initial placement and routing solution for each benchmark; whereas, for the largest device (Synthetic 6), the respective runtimes of the seam carving methods were 1.13%, 81.79%, and 97.61% greater than that of the initial placement and routing runtime. Altogether, this suggests favorable trends in terms of scalability proportional to device complexity.

Planar placement and routing is an efficient greedy heuristic, and thereby is unlikely to yield optimal results in terms of

area utilization. Consequently, many opportunities exist to discover and carve new seams, which increases the respective runtimes of the three seam carving methods. A longer-running initial placement and routing technique, which produces better quality results, is likely to yield fewer seams that can be discovered via post-processing, and, as a result, is likely to yield faster convergence times for the seam carving methods.

Routing graph construction dominates the runtime of NLSC and NRSC, but becomes amortized as the number of seams increases. This overhead could be further reduced if the same data structure was shared between the initial placer/router and the seam-carving post-processor. We did not explore this optimization because our objective was to make the seam carving post-processing algorithms wholly independent from the algorithm(s) used to for placement and routing.

#### E. Limitations and Potential Extensions

Prior work on both semiconductor [81] and FPGA [82] routing has shown that the result quality for algorithms that route one net at a time can be highly sensitive to the order in which nets are chosen for routing. As the techniques introduced in this paper identify and carve one seam at a time, they are likewise susceptible to ordering issues involving the identification and selection of seams to carve. Further, it is expected that the quality of the initial placement and routing result will impact the optimality of the overall carving outcome; initial solutions that are close to optimal are more likely to have fewer carvable seams, which will limit the effectiveness of post-processing. Future work on NLSC and NRSC may consider stochastic searches of potential paths to identify the maximum number of carvable seams; more aggressive approaches could also consider rip-up and re-route of routed nets with the objective of increasing the number of carvable seams relative to the initial start in position.

## VIII. CONCLUSION & FUTURE WORK

Seam carving was originally developed to reduce the size of images while minimizing loss in content quality. In this paper, we adopted the basic premise to eliminate unused space and reduce channel length in microfluidic chips. This approach to seam carving is likely to have applications in semiconductor technologies, such as printed circuit boards (PCBs). Future research could generalize seam carving to multi-layer microfluidic or semiconductor devices. Another topic work exploring is to study seam carving in a context in which certain channels or wires have an equal-length routing constraint; under this model, any seam (set or set of seams) must remove an equal amount of length from all channels or wires that are constrained to have equal length; otherwise the equal length constraint would be violated. A much greater challenge would be to apply seam carving to semiconductor VLSI chips, which combined a mixed or standard cells and IP blocks, along with multiple layers of metal for routing, including a clock tree. It would be both challenging and interesting to see if a generalized form of seam carving could be effective in this context.

## ACKNOWLEDGEMENT

This work was supported in part by NSF Awards #1351115, #1740052, #1910878, and #2019362.

## REFERENCES

- [1] T. Tseng, M. Li, B. Li, T. Ho, and U. 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*, 2016, pp. 147:1–147:6. [Online]. Available: <http://doi.acm.org/10.1145/2897937.2897997>
- [2] T. Tseng, M. Li, D. N. Freitas, T. McAuley, B. Li, T. Ho, I. E. Araci, and U. Schlichtmann, "Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1588–1601, 2018. [Online]. Available: <https://doi.org/10.1109/TCAD.2017.2760628>
- [3] A. Grimmer, Q. Wang, H. Yao, T. Ho, and R. 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*, pp. 530–535. [Online]. Available: <https://doi.org/10.1109/ASPDAC.2017.7858377>
- [4] T. Tseng, M. Li, D. N. Freitas, A. Mongersun, I. E. Araci, T. Ho, and U. 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*, 2018, pp. 163:1–163:6. [Online]. Available: <https://doi.org/10.1145/3195970.3196011>
- [5] J. McDaniel, B. Crites, P. Brisk, and W. H. Grover, "Flow-layer physical design for microchips based on monolithic membrane valves," *IEEE Design & Test*, vol. 32, no. 6, pp. 51–59, 2015. [Online]. Available: <http://dx.doi.org/10.1109/MDAT.2015.2459699>
- [6] B. Crites, K. Kong, and P. Brisk, "Diagonal component expansion for flow-layer placement of flow-based microfluidic biochips," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, p. 126, 2017.
- [7] —, "Directed placement for mvlsi devices," *J. Emerg. Technol. Comput. Syst.*, vol. 16, no. 2, pp. 14:1–14:26, Dec. 2019. [Online]. Available: <http://doi.acm.org/10.1145/3369585>
- [8] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, p. 10, 2007. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276390>
- [9] B. Crites, K. Kong, and P. Brisk, "Reducing microfluidic very large scale integration (mvlsi) chip area by seam carving," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 2017, pp. 459–462.
- [10] B. Crites, R. Sanka, J. Lippai, J. McDaniel, P. Brisk, and D. Densmore, "Parchmint: A microfluidics benchmark suite," in *2018 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 2018, pp. 78–79.
- [11] R. Sanka, B. Crites, J. McDaniel, P. Brisk, and D. Densmore, "Specification, integration, and benchmarking of continuous flow microfluidic devices: Invited paper," in *Proceedings of the International Conference on Computer-Aided Design, ICCAD 2019, Westminster, CO, USA, November 4-7, 2019*, D. Z. Pan, Ed. ACM, 2019, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/ICCAD45719.2019.8942171>
- [12] Y. Xia and G. M. Whitesides, "Soft lithography," *Annual Review of Materials Science*, vol. 28, no. 1, pp. 153–184, 1998. [Online]. Available: <https://doi.org/10.1146/annurev.matsci.28.1.153>
- [13] Y. Xia, E. Kim, X.-M. Zhao, J. A. Rogers, M. Prentiss, and G. M. Whitesides, "Complex optical surfaces formed by replica molding against elastomeric masters," *Science*, vol. 273, no. 5273, pp. 347–349, 1996. [Online]. Available: <https://science.sciencemag.org/content/273/5273/347>
- [14] X.-M. Zhao, A. Stoddart, S. P. Smith, E. Kim, Y. Xia, M. G. Prentiss, and G. M. Whitesides, "Fabrication of single-mode polymeric waveguides using micromolding in capillaries," *Advanced Materials*, vol. 8, pp. 420–424, 1996, 500.
- [15] J. A. Rogers, R. J. Jackman, and G. M. Whitesides, "Constructing single- and multiple-helical microcoils and characterizing their performance as components of microinductors and microelectromagnets," *Journal of Microelectromechanical Systems*, vol. 6, no. 3, pp. 184–192, Sep. 1997.
- [16] R. J. Jackman, S. T. Brittain, A. Adams, M. G. Prentiss, and G. M. Whitesides, "Design and fabrication of topologically complex, three-dimensional microstructures," *Science*, vol. 280, no. 5372, pp. 2089–2091, 1998. [Online]. Available: <https://science.sciencemag.org/content/280/5372/2089>
- [17] J. R. Anderson, D. T. Chiu, R. J. Jackman, O. Cherniavskaya, J. C. McDonald, H. Wu, S. H. Whitesides, and G. M. Whitesides, "Fabrication of topologically complex three-dimensional microfluidic systems in pdms by rapid prototyping," *Analytical Chemistry*, vol. 72, no. 14, pp. 3158–3164, 2000, pMID: 10939381. [Online]. Available: <https://doi.org/10.1021/ac9912294>
- [18] P. Yang, †. G. Wirnsberger, H. C. Huang, S. R. Cordero, M. D. McGehee, B. Scott, T. Deng, G. M. Whitesides, B. F. Chmelka, S. K. Buratto, and G. D. Stucky, "Mirrorless lasing from mesostructured waveguides patterned by soft lithography," *Science*, vol. 287, no. 5452, pp. 465–467, 2000. [Online]. Available: <https://science.sciencemag.org/content/287/5452/465>
- [19] C. S. Effenhauser, G. J. M. Bruin, A. Paulus, and M. Ehrat, "Integrated capillary electrophoresis on flexible silicone microdevices: Analysis of dna restriction fragments and detection of single dna molecules on microchips," *Analytical Chemistry*, vol. 69, no. 17, pp. 3451–3457, 1997, pMID: 21639267. [Online]. Available: <https://doi.org/10.1021/ac9703919>
- [20] D. C. Duffy, J. C. McDonald, O. J. A. Schueller, and G. M. Whitesides, "Rapid prototyping of microfluidic systems in poly(dimethylsiloxane)," *Analytical Chemistry*, vol. 70, no. 23, pp. 4974–4984, 1998, pMID: 21644679. [Online]. Available: <https://doi.org/10.1021/ac980656z>
- [21] H.-P. Chou, C. Spence, A. Scherer, and S. R. Quake, "Microfabricated devices for sizing DNA and sorting cells," in *Micro- and Nanofabricated Structures and Devices for Biomedical Environmental Applications*, P. L. Gourley, Ed., vol. 3258, International Society for Optics and Photonics. SPIE, 1998, pp. 181 – 187. [Online]. Available: <https://doi.org/10.1117/12.304378>
- [22] A. Y. Fu, C. Spence, A. Scherer, F. H. Arnold, and S. R. Quake, "A microfabricated fluorescence-activated cell sorter," *Nature Biotechnology*, vol. 17, no. 11, pp. 1109–1111, 1999.
- [23] S. R. Quake and A. Scherer, "From micro- to nanofabrication with soft materials," *Science*, vol. 290, no. 5496, pp. 1536–1540, 2000. [Online]. Available: <https://science.sciencemag.org/content/290/5496/1536>
- [24] G. M. Whitesides, E. Ostuni, S. Takayama, X. Jiang, and D. E. Ingber, "Soft lithography in biology and biochemistry," *Annual Review of Biomedical Engineering*, vol. 3, no. 1, pp. 335–373, 2001, pMID: 11447067. [Online]. Available: <https://doi.org/10.1146/annurev.bioeng.3.1.335>
- [25] S. Quake, "Solving the tyranny of pipetting," 2018.
- [26] W. H. Grover, A. M. Skelley, C. N. Liu, E. T. Lagally, and R. A. Mathies, "Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices," *Sensors and Actuators B: Chemical*, vol. 89, no. 3, pp. 315 – 323, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925400502004689>
- [27] W. H. Grover, R. H. C. Ivester, E. C. Jensen, and R. A. Mathies, "Development and multiplexed control of latching pneumatic valves using microfluidic logical structures," *Lab Chip*, vol. 6, pp. 623–631, 2006. [Online]. Available: <http://dx.doi.org/10.1039/B518362F>
- [28] M. A. Unger, H.-P. Chou, T. Thorsen, A. Scherer, and S. R. Quake, "Monolithic microfabricated valves and pumps by multilayer soft lithography," *Science*, vol. 288, no. 5463, pp. 113–116, 2000. [Online]. Available: <http://science.sciencemag.org/content/288/5463/113>
- [29] H.-P. Chou, M. A. Unger, and S. R. Quake, "A microfabricated rotary pump," *Biomedical Microdevices*, vol. 3, no. 4, pp. 323–330, 2001. [Online]. Available: <https://doi.org/10.1023/A:1012412916446>
- [30] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002. [Online]. Available: <https://science.sciencemag.org/content/298/5593/580>
- [31] J. P. Urbanski, W. Thies, C. Rhodes, S. Amarasinghe, and T. Thorsen, "Digital microfluidics using soft lithography," *Lab Chip*, vol. 6, pp. 96–104, 2006. [Online]. Available: <http://dx.doi.org/10.1039/B510127A>
- [32] J. W. Hong and S. R. Quake, "Integrated nanoliter systems," *Nature Biotechnology*, vol. 21, pp. 1179–1183, 2003. [Online]. Available: <https://doi.org/10.1038/nbt871>
- [33] J. Melin and S. R. Quake, "Microfluidic large-scale integration: The evolution of design rules for biological automation," *Annual Review of Biophysics and Biomolecular Structure*, vol. 36, no. 1, pp. 213–231, 2007, pMID: 17269901. [Online]. Available: <http://dx.doi.org/10.1146/annurev.biophys.36.040306.132646>
- [34] I. E. Araci, P. Pop, and K. Chakrabarty, "Microfluidic very large-scale integration for biochips: Technology, testing and fault-tolerant design," in *20th IEEE European Test Symposium, ETS 2015, Cluj-Napoca, Romania, 25-29 May, 2015*, 2012, pp. 1–8. [Online]. Available: <http://dx.doi.org/10.1109/ETS.2015.7138736>
- [35] C. L. Hansen, E. Skordalakes, J. M. Berger, and S. R. Quake, "A robust and scalable microfluidic metering method that allows protein

- crystal growth by free interface diffusion,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 26, pp. 16531–16536, 2002. [Online]. Available: <https://www.pnas.org/content/99/26/16531>
- [36] J. Liu, C. Hansen, and S. R. Quake, “Solving the “world-to-chip” interface problem with a microfluidic matrix,” *Analytical Chemistry*, vol. 75, no. 18, pp. 4718–4723, 2003, pMID: 14674446. [Online]. Available: <https://doi.org/10.1021/ac0346407>
- [37] J. W. Hong, V. Studer, G. Hang, W. F. Anderson, and S. R. Quake, “A nanoliter-scale nucleic acid processor with parallel architecture,” *Nature Biotechnology*, vol. 22, no. 4, pp. 435–439, 2004. [Online]. Available: <https://doi.org/10.1038/nbt951>
- [38] C. L. Hansen, M. O. A. Sommer, and S. R. Quake, “Systematic investigation of protein phase behavior with a microfluidic formulation,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 40, pp. 14431–14436, 2004. [Online]. Available: <https://www.pnas.org/content/101/40/14431>
- [39] F. K. Balagaddé, L. You, C. L. Hansen, F. H. Arnold, and S. R. Quake, “Long-term monitoring of bacteria undergoing programmed population control in a microchemostat,” *Science*, vol. 309, no. 5731, pp. 137–140, 2005. [Online]. Available: <https://science.sciencemag.org/content/309/5731/137>
- [40] J. S. Marcus, W. F. Anderson, and S. R. Quake, “Microfluidic single-cell mrna isolation and analysis,” *Analytical Chemistry*, vol. 78, no. 9, pp. 3084–3089, 2006, pMID: 16642997. [Online]. Available: <https://doi.org/10.1021/ac0519460>
- [41] C. L. Hansen, S. Classen, J. M. Berger, and S. R. Quake, “A microfluidic device for kinetic optimization of protein crystallization and in situ structure determination,” *Journal of the American Chemical Society*, vol. 128, no. 10, pp. 3142–3143, 2006, pMID: 16522084. [Online]. Available: <https://doi.org/10.1021/ja0576637>
- [42] D. Falconnet, A. Niemistö, R. J. Taylor, M. Rivicova, T. Galitski, I. Shmulevich, and C. L. Hansen, “High-throughput tracking of single yeast cells in a microfluidic imaging matrix,” *Lab Chip*, vol. 11, pp. 466–473, 2011. [Online]. Available: <http://dx.doi.org/10.1039/C0LC000228C>
- [43] K. Leung, H. Zahn, T. Leaver, K. M. Konwar, N. W. Hanson, A. P. Pagé, C.-C. Lo, P. S. Chain, S. J. Hallam, and C. L. Hansen, “A programmable droplet-based microfluidic device applied to multiparameter analysis of single microbes and microbial communities,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 20, pp. 7665–7670, 2012. [Online]. Available: <https://www.pnas.org/content/109/20/7665>
- [44] A. R. Wu, T. L. Kawahara, N. A. Rapicavoli, J. v. Riggelen, E. H. Shroff, L. Xu, D. W. Felsler, H. Y. Chang, and S. R. Quake, “High throughput automated chromatin immunoprecipitation as a platform for drug screening and antibody validation,” *Lab Chip*, vol. 12, pp. 2190–2198, 2012. [Online]. Available: <http://dx.doi.org/10.1039/C2LC21290K>
- [45] J. Huft, C. A. Haynes, and C. L. Hansen, “Microfluidic integration of parallel solid-phase liquid chromatography,” *Analytical Chemistry*, vol. 85, no. 5, pp. 2999–3005, 2013, pMID: 23384109. [Online]. Available: <https://doi.org/10.1021/ac400163u>
- [46] F. Piraino, F. Volpetti, C. Watson, and S. J. Maerkl, “A digital–analog microfluidic platform for patient-centric multiplexed biomarker diagnostics of ultralow volume samples,” *ACS Nano*, vol. 10, no. 1, pp. 1699–1710, 2016, pMID: 26741022. [Online]. Available: <https://doi.org/10.1021/acsnano.5b07939>
- [47] I. E. Araci, M. Robles, and S. R. Quake, “A reusable microfluidic device provides continuous measurement capability and improves the detection limit of digital biology,” *Lab Chip*, vol. 16, pp. 1573–1578, 2016. [Online]. Available: <http://dx.doi.org/10.1039/C6LC00194G>
- [48] R. H. W. Lam, X. Cui, W. Guo, and T. Thorsen, “High-throughput dental biofilm growth analysis for multiparametric microenvironmental biochemical conditions using microfluidics,” *Lab Chip*, vol. 16, pp. 1652–1662, 2016. [Online]. Available: <http://dx.doi.org/10.1039/C6LC00072J>
- [49] L. M. Fidalgo and S. J. Maerkl, “A software-programmable microfluidic device for automated biology,” *Lab Chip*, vol. 11, pp. 1612–1619, 2011. [Online]. Available: <http://dx.doi.org/10.1039/C0LC00537A>
- [50] E. C. Jensen, W. H. Grover, and R. A. Mathies, “Micropneumatic digital logic structures for integrated microdevice computation and control,” *Journal of Microelectromechanical Systems*, vol. 16, no. 6, pp. 1378–1385, Dec 2007.
- [51] A. M. Skelley, J. R. Scherer, A. D. Aubrey, W. H. Grover, R. H. C. Ivester, P. Ehrenfreund, F. J. Grunthaler, J. L. Bada, and R. A. Mathies, “Development and evaluation of a microdevice for amino acid biomarker detection and analysis on mars,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 4, pp. 1041–1046, 2005. [Online]. Available: <https://www.pnas.org/content/102/4/1041>
- [52] J. Kim, E. C. Jensen, M. Megens, B. Boser, and R. A. Mathies, “Integrated microfluidic bioprocessor for solid phase capture immunoassays,” *Lab Chip*, vol. 11, pp. 3106–3112, 2011. [Online]. Available: <http://dx.doi.org/10.1039/C1LC20407F>
- [53] K. Du, H. Cai, M. Park, T. Wall, M. Stott, K. Alfson, A. Griffiths, R. Carrion, J. Patterson, A. Hawkins, H. Schmidt, and R. Mathies, “Multiplexed efficient on-chip sample preparation and sensitive amplification-free detection of ebola virus,” *Biosensors and Bioelectronics*, vol. 91, pp. 489 – 496, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0956566316313197>
- [54] K. Du, M. Park, A. Griffiths, R. Carrion, J. Patterson, H. Schmidt, and R. Mathies, “Microfluidic system for detection of viral rna in blood using a barcode fluorescence reporter and a photocleavable capture probe,” *Analytical Chemistry*, vol. 89, no. 22, pp. 12433–12440, 2017, pMID: 29073356. [Online]. Available: <https://doi.org/10.1021/acs.analchem.7b03527>
- [55] E. C. Jensen, B. P. Bhat, and R. A. Mathies, “A digital microfluidic platform for the automation of quantitative biomolecular assays,” *Lab Chip*, vol. 10, pp. 685–691, 2010. [Online]. Available: <http://dx.doi.org/10.1039/B920124F>
- [56] G. Linshiz, E. Jensen, N. Stawski, C. Bi, N. Elsbree, H. Jiao, J. Kim, R. Mathies, J. D. Keasling, and N. J. Hillson, “End-to-end automated microfluidic platform for synthetic biology: from design to functional analysis,” *Journal of Biological Engineering*, vol. 10, no. 1, 2016. [Online]. Available: <https://doi.org/10.1186/s13036-016-0024-5>
- [57] L. Wolf, M. Guttman, and D. Cohen-Or, “Non-homogeneous content-driven video-retargeting,” in *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007, 2007*, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/ICCV.2007.4409010>
- [58] M. Rubinstein, A. Shamir, and S. Avidan, “Improved seam carving for video retargeting,” *ACM Trans. Graph.*, vol. 27, no. 3, p. 16, 2008. [Online]. Available: <https://doi.org/10.1145/1360612.1360615>
- [59] Y. Wang, C. Tai, O. Sorkine, and T. Lee, “Optimized scale-and-stretch for image resizing,” *ACM Trans. Graph.*, vol. 27, no. 5, p. 118, 2008. [Online]. Available: <https://doi.org/10.1145/1409060.1409071>
- [60] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “Summarizing visual data using bidirectional similarity,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA, 2008*. [Online]. Available: <https://doi.org/10.1109/CVPR.2008.4587842>
- [61] Y. Pritch, E. Kav-Venaki, and S. Peleg, “Shift-map image editing,” in *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009, 2009*, pp. 151–158. [Online]. Available: <https://doi.org/10.1109/ICCV.2009.5459159>
- [62] W. H. Minhass, P. Pop, and J. Madsen, “System-level modeling and synthesis of flow-based microfluidic biochips,” in *Proceedings of the 14th International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, CASES 2011, part of the Seventh Embedded Systems Week, ESWeek 2011, Taipei, Taiwan, October 9-14, 2011, 2011*, pp. 225–234. [Online]. Available: <https://doi.org/10.1145/2038698.2038733>
- [63] W. H. Minhass, P. Pop, J. Madsen, and F. S. Blaga, “Architectural synthesis of flow-based microfluidic large-scale integration biochips,” in *Proceedings of the 15th International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, CASES 2012, Tampere, Finland, October 7-12, 2012, 2012*, pp. 181–190. [Online]. Available: <http://doi.acm.org/10.1145/2380403.2380437>
- [64] H. Yao, Q. Wang, Y. Ru, Y. Cai, and T. Ho, “Integrated flow-control codesign methodology for flow-based microfluidic biochips,” *IEEE Design & Test*, vol. 32, no. 6, pp. 60–68, 2015. [Online]. Available: <http://dx.doi.org/10.1109/MDAT.2015.2449180>
- [65] C. Lin, C. Liu, I. Chen, D. T. Lee, and T. Ho, “An efficient bi-criteria flow channel routing algorithm for flow-based microfluidic biochips,” in *The 51st Annual Design Automation Conference 2014, DAC '14, San Francisco, CA, USA, June 1-5, 2014, 2014*, pp. 141:1–141:6. [Online]. Available: <https://doi.org/10.1145/2593069.2593084>
- [66] J. McDaniel, B. Parker, and P. Brisk, “Simulated annealing-based placement for microfluidic large scale integration (mlsi) chips,” in *22nd International Conference on Very Large Scale Integration, VLSI-SoC, Playa del Carmen, Mexico, October 6-8, 2014, 2014*, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/VLSI-SoC.2014.7004170>
- [67] Q. Wang, Y. Ru, H. Yao, T. Ho, and Y. Cai, “Sequence-pair-based placement and routing for flow-based microfluidic biochips,” in *21st Asia and South Pacific Design Automation Conference, ASP-DAC 2016, Macao, Macao, January 25-28, 2016, 2016*, pp. 587–592. [Online]. Available: <https://doi.org/10.1109/ASPAC.2016.7428075>

- [68] J. Wu, K. S. Li, J. Li, S. Wang, and T. Ho, "SOLAR: simultaneous optimization of control-layer pins placement and channel routing in flow-based microfluidic biochips," in *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, April 16-19, 2018*, 2018, pp. 1–4. [Online]. Available: <https://doi.org/10.1109/VLSI-DAT.2018.8373234>
- [69] Q. Wang, H. Zou, H. Yao, T. Ho, R. Wille, and Y. Cai, "Physical co-design of flow and control layers for flow-based microfluidic biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 6, pp. 1157–1170, 2018. [Online]. Available: <https://doi.org/10.1109/TCAD.2017.2748003>
- [70] U. Lauther, "A data structure for gridless routing," in *Proceedings of the 17th Design Automation Conference, DAC '80, Minneapolis, Minnesota, USA, June 23-25, 1980*, E. B. H. Jr., Ed. ACM/IEEE, 1980, pp. 603–609. [Online]. Available: <https://doi.org/10.1145/800139.804593>
- [71] J. Dion and L. Monier, "Contour: A tile-based gridless router," HP Labs, , 1995.
- [72] J. Cong, J. Fang, and K. Khoo, "Dune-a multilayer gridless routing system," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 20, no. 5, pp. 633–647, 2001. [Online]. Available: <https://doi.org/10.1109/43.920694>
- [73] J. Cong, J. Fang, M. Xie, and Y. Zhang, "Mars-a multilevel full-chip gridless routing system," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 382–394, 2005. [Online]. Available: <https://doi.org/10.1109/TCAD.2004.842803>
- [74] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electronic Computers*, vol. 10, no. 3, pp. 346–365, 1961. [Online]. Available: <http://dx.doi.org/10.1109/TEC.1961.5219222>
- [75] "Lemon graph library," <https://lemon.cs.elte.hu/trac/lemon>, accessed: 2020-02-15.
- [76] B. Li, L. Li, A. Guan, Q. Dong, K. Ruan, R. Hu, and Z. Li, "A smartphone controlled handheld microfluidic liquid handling system," *Lab Chip*, vol. 14, pp. 4085–4092, 2014. [Online]. Available: <http://dx.doi.org/10.1039/C4LC00227J>
- [77] M. Rhee and M. A. Burns, "Microfluidic assembly blocks," *Lab Chip*, vol. 8, pp. 1365–1373, 2008. [Online]. Available: <http://dx.doi.org/10.1039/B805137B>
- [78] A. R. Wu, J. B. Hiatt, R. Lu, J. L. Attema, N. A. Lobo, I. L. Weissman, M. F. Clarke, and S. R. Quake, "Automated microfluidic chromatin immunoprecipitation from 2,000 cells," *Lab on a Chip*, vol. 9, no. 10, pp. 1365–1370, 2009.
- [79] A. M. Amin, R. Thakur, S. Madren, H.-S. Chuang, M. Thottethodi, T. N. Vijaykumar, S. T. Wereley, and S. C. Jacobson, "Software-programmable continuous-flow multi-purpose lab-on-a-chip," *Microfluidics and nanofluidics*, vol. 15, no. 5, pp. 647–659, Nov. 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3889662/>
- [80] W. H. Minhass, P. Pop, and J. Madsen, "System-level modeling and synthesis of flow-based microfluidic biochips," in *Proceedings of the 14th international conference on Compilers, architectures and synthesis for embedded systems*. ACM, 2011, pp. 225–234.
- [81] L. C. Abel, "On the ordering of connections for automatic wire routing," *IEEE Trans. Computers*, vol. 21, no. 11, pp. 1227–1233, 1972. [Online]. Available: <https://doi.org/10.1109/T-C.1972.223482>
- [82] R. Rubin and A. DeHon, "Timing-driven pathfinder pathology and remediation: quantifying and reducing delay noise in vpr-pathfinder," in *Proceedings of the ACM/SIGDA 19th International Symposium on Field Programmable Gate Arrays, FPGA 2011, Monterey, California, USA, February 27, March 1, 2011*, J. Wawrzynek and K. Compton, Eds. ACM, 2011, pp. 173–176. [Online]. Available: <https://doi.org/10.1145/1950413.1950447>



**Brian Crites** Brian Crites received the B.S. degree in computer engineering in 2012 as well as M.S., and Ph.D. degrees in computer science in 2014 and 2018 (respectively) from the University of California, Riverside, Riverside, CA, USA. From 2018 to 2019, he was an Assistant Project Scientist working on physical design and post-processing algorithms for integrated microfluidic devices. He is currently a software engineer at PiñataFarms.



**Cody Falzone** Cody Falzone received the B.S. degree in computer engineering from the University of California, Riverside, Riverside, CA, USA in 2018. While at UCR, he worked under the supervision of Dr. Philip Brisk, Dr. Brian Crites, and Dr. Jeffrey McDaniel on physical design post-processing algorithms for integrated microfluidic devices. He is currently a software engineer at Glidewell Dental developing CAD/CAM software.



**Tristan Lopez** Tristan Lopez received the B.S. degree in Electrical Engineering in 2020 from the University of California, Riverside, Riverside, CA, USA. While at UCR, he worked under the supervision of Dr. Philip Brisk and Dr. Brian Crites on physical design post-processing algorithms for integrated microfluidic devices.



**Karen Kong** Karen Kong received the B.S. degree in computer science with the University of California, Riverside, Riverside, CA, USA in 2018. While at UCR, she worked under the supervision of Dr. Philip Brisk, Dr. Brian Crites, and Dr. Jeffrey McDaniel on physical design algorithms and benchmark suite development for integrated microfluidic devices. She is currently a software engineer at Facebook.



**Philip Brisk** Philip Brisk received the B.S., M.S., and Ph.D. degrees from the University of California at Los Angeles, Los Angeles, CA, USA, in 2002, 2003, and 2006, respectively, all in computer science.

From 2006 to 2009, he was a Post-Doctoral Scholar with the Processor Architecture Laboratory, School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is presently a Professor with the Department of Computer Science and Engineering, University of California at Riverside, Riverside, CA, USA. His current research interests include microfluidics, FPGAs, programming languages and compilers, and application-specific processors.

Dr. Brisk was a recipient of the Best Paper Award at CASES 2007, FPL 2009, and ReConfig 2019. He has been a Program Committee member for several conferences and workshops, including DAC, ASPDACC, DATE, VLSI-SoC, FPL, and FPT. He has been the General (Co)-Chair of the IEEE SIES 2009, the IEEE SASP 2010, and IWLS 2011, and the Program (Co)-Chair of the IEEE SASP 2011, IWLS 2012, ARC 2013, and FPL 2016. He is an Associate Editor of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and of Integration: The VLSI Journal.