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

Brian Crites, Karen Kong, Philip Brisk
University of California, Riverside
900 University Ave
Riverside, CA
bcrit001@ucr.edu, kkong006@ucr.edu, philip@cs.ucr.edu

ABSTRACT

This paper introduces a technique based on seam carving to reduce the area of microfluidic very large scale integration (mVLSI) chips. Seam carving repeatedly identifies small slices of the device that can be safely removed (carved) and patched without adversely affecting device functionality. Using non-linear seam carving we achieve an average improvement of $4.28x$ in area utilization and an average reduction in fluid routing channel length of 53%.

Keywords

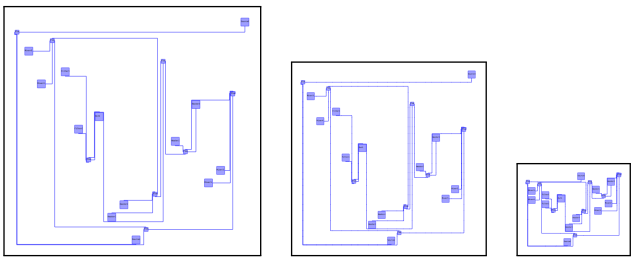
Microfluidics, Seam Carving, mVLSI

1. 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.

The primary component of modern LoCs is the microvalve. This device is fabricated using multi-layer soft lithography with two layers of flexible polymer, polydimethylsiloxane (PDMS), mounted on top of a rigid substrate, typically a glass slide. The two logical layers of these devices are the "flow layer," which transports biological fluids of an assay, and the "control layer," which provides actuation capabilities. A microvalve forms where a control channel crosses a flow channel. By default, all microvalves are open; pressurizing a control channel closes all of the microvalves that it drives. Components, such as pumps, mixers, switches, etc. can be built from microvalves [6].

As *microfluidic Very Large Scale Integration (mVLSI)* densities increase [1], designers will require CAD tools to cope with increasing device complexity. Like integrated circuits, an mVLSI chip can be viewed as a netlist, which must be placed and routed, with the restriction that each layer must be planar. Optimal mVLSI physical design has been achieved using integer linear programming [10], however, it is unlikely that this approach will scale to large devices. Meanwhile, tractable heuristics based on planar graph embedding [5] yield solutions that are far from optimal.



(a) (b) (c)

Figure 1: (a) Shows the benchmarks Synthetic 1 after the baseline placement and routing [5] has completed. (b) is the same benchmark after linear seam carving has been applied, while (c) is after non-linear seam carving has been applied.

This paper adapts seam carving [2], an image size reduction technique, to improve the area utilization of low-quality mVLSI layouts. The basic premise is to identify seams (paths) through the chip which can be removed without adversely affecting device functionality, shortening fluid channels that may be cut. Fig. 1 shows a motivating example. Fig. 1a shows a low quality initial layout [5]. Fig. 1b shows an improved layout, which was derived using linear seam carving, which we introduce in Section 4; 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 with a more aggressive technique, nonlinear seam carving, which we introduce in Section 5; non-linear seam carving improves area utilization by $4.28x$ on average, while reducing average fluid routing channel length by 53%

2. RELATED WORK

2.1 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 graph, where each vertex represents a pixel and each vertex's weight represents its relative importance to image quality [2]. Seam carving then finds the lowest-cost path from one perimeter edge to its opposite and removes it from the image. The process repeats until the desired reduction in size is achieved.

GLSVLSI '17, May 10-12, 2017, Banff, AB, Canada

DOI: http://dx.doi.org/10.1145/3060403.3060461

2.2 mVLSI Placement

Seam carving can reduce the area of mVLSI chips designed manually, or laid out using sub-optimal heuristic methods such as simulated annealing [7], incremental cluster expansion [9] and extensions to planar graph embedding [5]. Seam carving will not be able to improve an optimal placement result [10] because the existence of a removable seam contradicts the optimality of the result.

3. PRELIMINARIES

The input to seam carving is a placed and routed mVLSI architecture $A = (C, R, n, m)$, where C is a set of placed components, R is a set of routed channel segments, and n and m are the respective height and width of the layout. We represent each microfluidic component $c_i = (x_i, y_i, w_i, h_i)$ using a bounding box: point (x_i, y_i) is the upper-left corner of the component, and h_i and w_i are its respective height and width. Each routed channel segment $r_i = (x_{i,t}, y_{i,t}, x_{i,l}, y_{i,l})$ is a straight-line connection between points $(x_{i,t}, y_{i,t})$ and $(x_{i,l}, y_{i,l})$; multiple segments may comprise a longer channel with twists and bends. The physical layout process may include an additional parameter, Δ, which adds white space around each component to improve routability.

In microfluidic devices all space in an architecture can be classified into three categories. Components, which have a fixed height and width; we assume that component dimensions are fixed by fluidic IP designers and cannot be reduced without adversely affecting chip functionality. Fluidic channels, which can be of any length as long as they provide a continuous flow of fluid between source and sink components; channel length can be reduced without altering chip functionality. Free space, which is superfluous, except for the buffer space surrounding each component.

A seam is a path through the architecture that connects one perimeter edge to its opposite and contains no points that are invalid for removal; this ensures that correct device functionality is maintained when the seam is removed. Invalid points include any part of a component (including its buffer space) or a switch at a channel intersection. In the latter case, removal of a switch would require the post-processor to re-place the switch and reroute its incident fluid channels accordingly; it is preferable to avoid this overhead. Valid points for removal include free space and channel segments that are not part of a switch and would not break the route connection between connected components.

4. LINEAR SEAM CARVING

Linear seam carving restricts seams to be horizontal or vertical straight lines that do not bend. Fig. 2a shows an example mVLSI chip with a loose placement and ample white space. Fig. 2b shows four horizontal seams, two of which intersect fluid channels in the center of the chip. Fig. 2c 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.

4.1 Seam Identification

Linear seam carving employs two Boolean arrays, B_x and B_y, which respectively 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 the component having i as the x-coordinate.

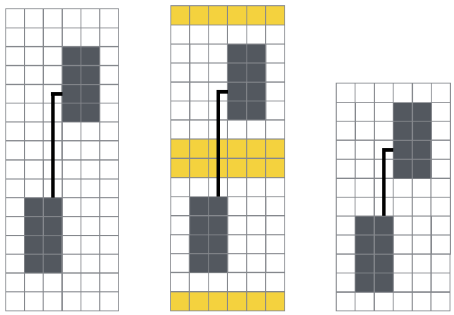


Figure 2: (a) A laid out mVLSI chip; (b) seam identification ($\Delta = 1$); (c) the chip after seam removal.

Both arrays are initialized to $B_x[0:m] = B_y[0:n] = True$.

The algorithm identifies vertical and horizontal seams for removal separately. To identify vertical seams, the algorithm iterates through all components $c_i \in C$ setting $B_x[x_i : x_i + w_i] = False$; this disallows any seam that cuts through a component. For each route $r_i \in R$ the algorithm sets $B_x[x_{i,t}] = False$ and $B_x[x_{i,l}] = False$ to disallow the removal of switches at channel intersections. 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.

Seams are permitted to cut through channel segments, effectively shortening them. If a channel segment of a prespecified length is required (e.g., to achieve a chemical separation), then it should be characterized as a component.

4.2 Seam Carving

Each index $j \in \{0, ..., m\}$ 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. For channel segments that cross the seam, the right endpoint is shifted left by one grid point. Seam carving cannot completely remove a channel because seams cannot contain channel endpoints. The final step is to reduce the length of the guide B_x by one grid point by setting $B_x[k] = B_x[k+1], j \leq k \leq m$, and decrementing m.

This process then repeats similarly for all vertical seams, $0 \leq j \leq n$ where $B_y[j]$ is $True$.

5. NON-LINEAR SEAM CARVING

Non-linear seam carving eliminates the restriction that seams are 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 removal and can lead to substantially smaller chip designs.

5.1 Seam Identification

Seam identification employs an $m \times n$ Boolean grid G to determine if a given point is a candidate for seam carving. All grid entries are initialized to $True$. For each component $c_i \in C$ at position (x_i, y_i) we set $G[j][k] = False$, $x_i \leq j \leq x_i + w_i, y_i \leq k \leq y_i + h_i$, rendering these points invalid for inclusion in a seam. For each routed channel segment $r_i \in R$ we set $G[x_{i,t}][y_{i,t}] = G[x_{i,l}][y_{i,l}] = False$ to disallow seam carving through switches at channel in-

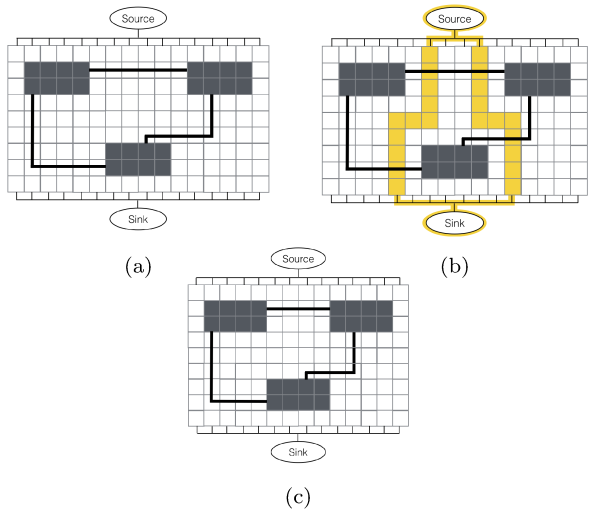


Figure 3: (a) A placed and routed mVLSI netlist (b) two nonlinear seams identified for removal ($\Delta = 1$); (c) the smaller chip after seam removal.

tersection points. A seam S is a collection of straight line segments $s_i = (a_i, b_i, c_i, d_i)$, where (a_i, b_i) and (c_i, d_i) are the (x, y)-coordinates of the two endpoints.

Non-linear seam carving retains the directional approach of its linear counterpart. Seams are first identified along the x-axis, with an artificial *source* connected to all grid positions $G[j][0], 0 \leq j \leq m$ and and artificial *sink* connected to all grid positions $G[j][n], 0 \leq j \leq m$, as shown in Fig. 3a. Lee's Algorithm [3] is repeatedly called to identify seams from *source* to *sink*, until no valid paths remain. Fig. 3b shows two non-linear seams, whose removal yields the smaller chip depicted in Fig. 3c. This process then repeats along the y-axis.

5.2 Perpendicular Channel Segments

Non-linear seam carving requires special handling of channel segments that run perpendicular to the carving direction. Without loss of generality, assume that we are carving in the y-direction and consider a horizontal channel segment r_i having $y_{i,t} = y_{i,l}$. A seam can be identified that cuts through r_i in such a way that its removal causes $y_{i,t} \neq y_{i,l}$; the updated channel would require a diagonal connection, or a small bend (necessitating three new channel segments), neither of which is problematic, per se.

Perpendicular carving, however, can cause a component to shift and collide with the perpendicular channel. For example, Fig. 4a shows a laid out mVLSI chip. Fig. 4b shows multiple seams that cross perpendicular channel segments; in Fig. 4c, carving these seams causes a component to shift and collide with the perpendicular channel. To prevent this, non-linear carving may not carve through perpendicular channel segments by setting $G[x_{i,t}][z] = False$ for $y_{i,t} < z < y_{i,l}$; Fig. 4d depicts a valid set of seams, and Fig. 4e shows the resulting collision-free mVLSI chip after carving.

5.3 Seam Carving

Non-linear seam carving must choose whether to move a component or channel segment endpoint based on the opposite axis along the seam. When carving along the x-axis, any component $c_i \in C$ that exists to the right of a seam

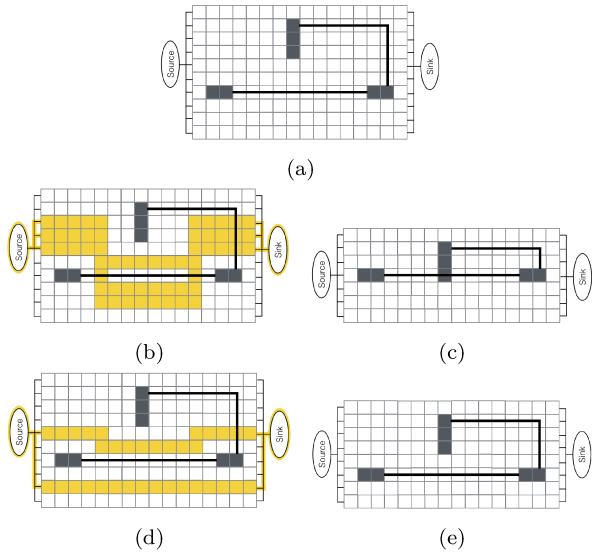


Figure 4: (a) A placed and routed mVLSI chip; (b) when carving along the y-axis ($\Delta = 1$), a set of non-linear seams are found that cross a perpendicular (horizontal) segment; (c) removal of the preceding seams yields an invalid layout; (d) non-linear seams are prevented from crossing the perpendicular segment; (e) removal of non-linear seams that do not cross the perpendicular segment yields a legal layout.

with $x_i > a_j$ between $b_j \leq y_i \leq d_j$ for any segment $s_j \in S$ will be shifted left to fill the the space that has been carved; to do this, set $x_i = x_i - 1$. All channel segments $r_i \in R$ with a source to the right of the seam with $x_{i,t} > a_j$ and $b_j \leq y_{i,t} \leq d_j$ will be shifted left to $x_{i,t} = x_{i,t} - 1$; all segments with a sink to the right of the seam with $x_{i,l} > a_j$ and $b_j \leq y_{i,l} \leq d_j$ will be shifted left to $x_{i,l} = x_{i,l} - 1$.

This process then repeats similarly along the y-axis.

6. EXPERIMENTAL RESULTS

Our Baseline algorithm is the mVLSI flow layer placer and router described by McDaniel et al. [5]. We implemented the Baseline algorithm in C++, along with linear and non-linear seam carving as post-processing steps. All experiments were run on a 2015 MacBook Pro Laptop with a 2.9GHz Processor and 8GB of RAM. For evaluation, we use a suite of nine benchmarks: AquaFlex-3b and AquaFlex-5a (proprietary netlists provided by Microfluidic Innovations LLC), a bead-based HIV1 immunoassay (Li et al. [4]), a molecular gradients generator (Rhee & Burns [8]), and five synthetic netlists. Our implementation uses a "unitless grid" with a standard buffer size $\Delta = 5$ for all benchmarks. We report the area utilization, (the percentage of the chip area consumed by components; Fig. 5), average channel routing length (Fig. 6), and the average algorithmic runtimes across five runs per algorithm/benchmark (Fig. 7).

Compared to the Baseline placement, linear seam carving marginally improved area utilization and average wirelength, while non-linear seam carving yielded far more significant improvements. The Baseline placer is ineffective because its underlying planar graph embedding algorithm does not try to minimize area, and further loses efficiency as vertices (points) are expanded into 2D components, which necessitates further shifting of components and re-routing of flow channels to eliminate overlap. Seam carving can effec-

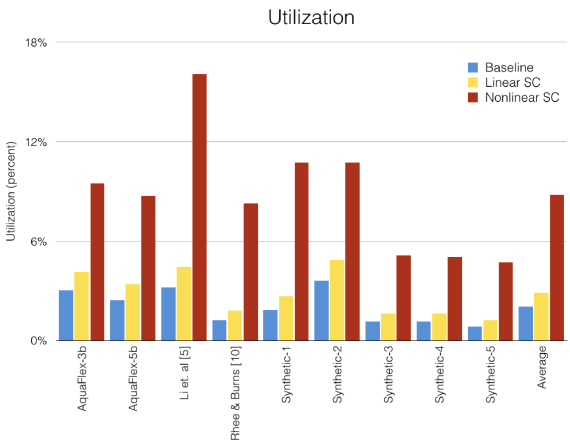


Figure 5: Area utilization (larger is better)

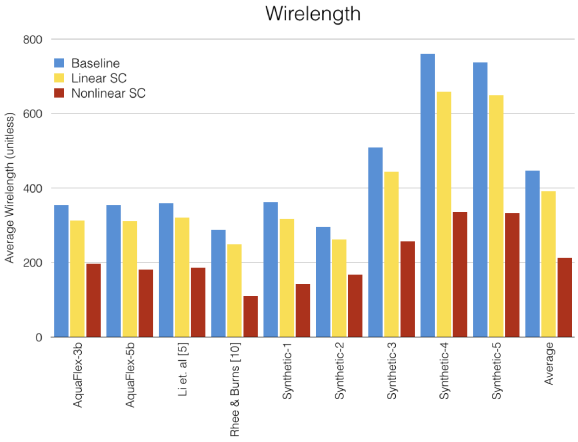


Figure 6: Average wirelength (smaller is better)

tively counteract these inefficiencies, and the results clearly show that there are far more non-linear seams available for removal than linear seams.

The runtimes reported in Fig. 7 include the Baseline placer in all cases. Although its effectiveness is limited, linear seam carving imposes negligible runtime overhead; in contrast, the runtime of non-linear seam carving is inversely proportional to the density of the design, and, as a post-processing step, it often runs longer than the Baseline placer (e.g., Synthetic-1 and -2).

7. CONCLUSION AND FUTURE WORK

Linear and non-linear seam carving can reduce the amount of unused space in an mVLSI chip. Linear seam carving is more conservative and runs faster, improving area utilization by by $1.4x$ and reducing the average wirelength by 13% on average; non-linear seam carving is far more aggressive, improving average area utilization by $4.28x$ while reducing average wirelength by 53%. Compared to the baseline placer however, this comes at a much greater runtime cost.

Acknowledgement

This work was support in part by NSF Awards #1351115, #1536026 and #1640757.

8. REFERENCES

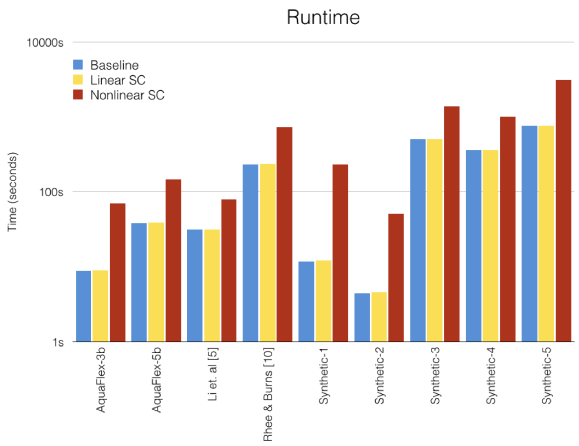


Figure 7: Algorithmic runtime in seconds (log. scale)

[1] Araci, I. E., Pop, P., and Chakrabarty, K. 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*, pp. 1–8.

[2] Avidan, S., and Shamir, A. Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10.

[3] Lee, C. Y. An algorithm for path connections and its applications. *IRE Trans. Electronic Computers 10*, 3 (1961), 346–365.

[4] Li, B., Li, L., Guan, A., Dong, Q., Ruan, K., Hu, R., and Li, Z. A smartphone controlled handheld microfluidic liquid handling system. *Lab Chip 14* (2014), 4085–4092.

[5] McDaniel, J., Crites, B., Brisk, P., and Grover, W. H. Flow-layer physical design for microchips based on monolithic membrane valves. *IEEE Design & Test 32*, 6 (2015), 51–59.

[6] Melin, J., and Quake, S. R. Microfluidic large-scale integration: The evolution of design rules for biological automation. *Annual Review of Biophysics and Biomolecular Structure 36*, 1 (2007), 213–231. PMID: 17269901.

[7] Minhass, W. H., Pop, P., Madsen, J., and Blaga, F. S. 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*, pp. 181–190.

[8] Rhee, M., and Burns, M. A. Microfluidic assembly blocks. *Lab Chip 8* (2008), 1365–1373.

[9] Tseng, K., You, S., Liou, J., and Ho, T. A top-down synthesis methodology for flow-based microfluidic biochips considering valve-switching minimization. In *International Symposium on Physical Design, ISPD'13, Stateline, NV, USA, March 24-27, 2013*, pp. 123–129.

[10] Tseng, T., Li, M., Li, B., Ho, T., and Schlichtmann, U. 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*, pp. 147:1–147:6.