THE CHIPS TO SYSTEMS CONFERENCE
61
SHAPING THE NEXT GENERATION OF ELECTRONICS
JUNE 23-27, 2024
MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Fracturing-aware Curvilinear ILT via Circular E-beam Mask Writer

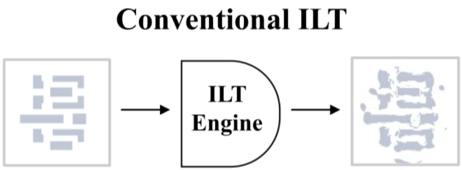
Xinyun Zhang, Su Zheng, Guojin Chen, Binwu Zhu, Hong Xu, Bei Yu

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Outline

Introduction

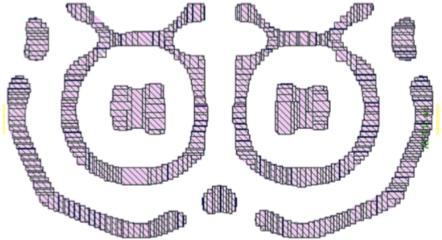
The promise of ILT



- ILT tends to generate curvilinear masks.
- Curvilinear masks achieve the best process window[1].

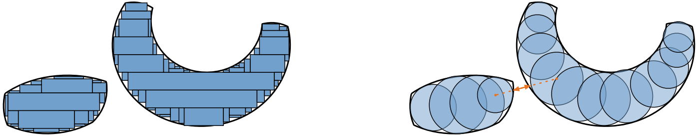
[1] Linyong Pang (2021). "Inverse lithography technology: 30 years from concept to practical, full-chip reality". In: *Journal of Micro/Nanopatterning, Materials, and Metrology*.

Roadblock to broad application of ILT



- Traditional variable-shaped beam (VSB) mask writers use rectlinear shapes to create mask shapes.
- The rectlinear fracturing requires many VSB shots.

A potential solution: Circular E-beam Mask Writers[2]



(a) Rectangular Fracturing   (b) Circular Fracturing

- Circular e-beam mask writers write main features as circles.
- It can significantly reduce the shot count.
- It is mask rule checking-friendly.

[2] Aki Fujimura et al. (2010). "Best depth of focus on 22-nm logic wafers with less shot count". In: *Photomask and Next-Generation Lithography Mask Technology XVII*.

Circular fracturing-aware OPC (CFAOPC)

Problem Definition

- Input: A target layout
- Output: A mask fractured by circles.
- Aim: Achieve the best mask performance while maintaining the minimal shot count (# of circles).

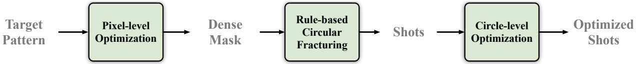
Key Differences:

- Consider OPC and fracturing in a unified perspective.
- The fracturing shapes are circles instead of rectangles.

Proposed Flow

- Rule-based method (CircleRule):

- Optimization-based method (CircleOpt):



CircleRule

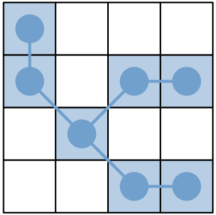
Rule-based Method: CircleRule

Overview:

- Sample shot (circle) centers in each feature
- Assign a radius to each shot

CircleRule: Circle Center Sampling

- Construct a skeleton graph
- Perform depth-first search (DFS), and sample points at a fixed sample rate.



Construction of the skeleton graph.

CircleRule: Radius selection

- Given a circle $C((x, y), r)$, the cover rate can be defined as $\frac{|C((x,y),r) \cap S|}{|C((x,y),r)|}$.
- Increase the radius until the cover rate reaches a threshold.

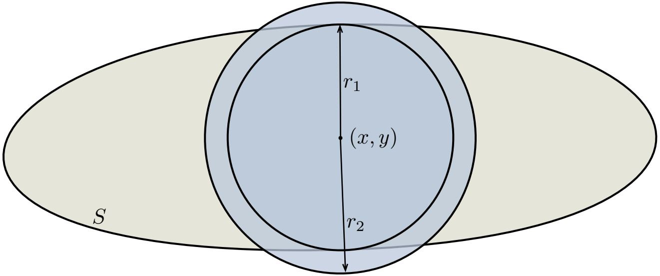
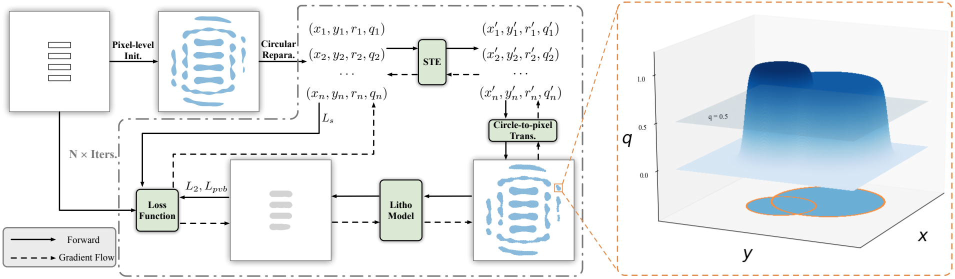


Illustration of the radius selection

CircleOpt

Optimization-based method: CircleOpt

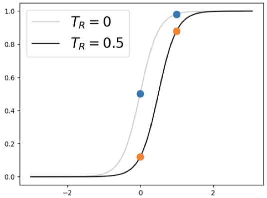


Overall flow of CircleOpt

- Two-stage optimization: Pixel-level initialization + Circle-level finetuning
- For circle-level optimization, we encode the circular constraints into ILT.

CircleOpt: Pixel-level Initialization

- The simplest pixel-level ILT[3].
- To generate masks with SRAFs, we use the shifted binary function[4].



The shifted binary function.

[3] Jhih-Rong Gao et al. (2014). "MOSAIC: Mask Optimizing Solution With Process Window Aware Inverse Correction". In: *DAC*.

[4] Shuyuan Sun et al. (2023). "Efficient ILT via Multi-level Lithography Simulation". In: *DAC*.

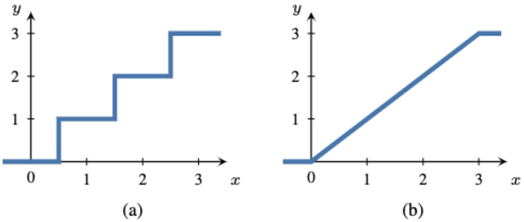
CircleOpt: Sparse Circular Reparameterization

- Similar to CircleRule, we fracture the initialized mask $\boldsymbol{M}$ into a set of sparse circular representations $\{(x_1, y_1, r_1, q_1), \cdots, (x_n, y_n, r_n, q_n)\}$.
- Our target becomes optimizing these **4n** variables for best mask performance and shot count.

CircleOpt: Differentiable circle-to-pixel transformation

- The coordinate x, y and the radius r must be integers in a dense mask.
- Use straight-through estimator (STE) to quantize x, y and r as

$$x_i' = \mathrm{STE}(x_i), y_i' = \mathrm{STE}(y_i), r_i' = \mathrm{STE}(r_i). \tag{1}$$



Visualization of the straight-through estimator. (a) STE forward; (b) STE backward.

CircleOpt: Differentiable circle-to-pixel transformation

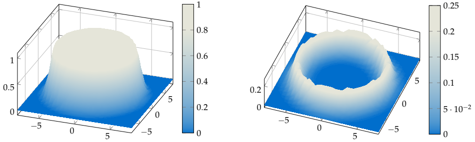
- We define the window function as

$$f_{x'_i,y'_i,r'_i}(x,y) = \frac{1}{1+e^{-\alpha(-\sqrt{(x-x'_i)^2+(y-y'_i)^2}+r'_i)}}, \tag{2}$$

$g(x) = x(1-x)$, and $h_{x'_i,y'_i,r'_i}(x,y) = (g \circ f_{x'_i,y'_i,r'_i})(x,y)$.

- Then, the dense mask can be obtained by

$$\bar{\boldsymbol{M}}(x,y) = \max_{i\in\{1,\cdots,n\}} \{q_i f_{x'_i,y'_i,r'_i}(x,y)\}. \tag{3}$$



(a) (b)

(a) Visualization of the circular window function $f_{0,0,6}(x,y)$; (b) Visualization of $h_{0,0,6}(x,y)$

CircleOpt: Differentiable circle-to-pixel transformation

- The gradient *w.r.t* the coordinate, the radius, and the activation can be represented as:

$$\frac{\partial \bar{\boldsymbol{M}}(x,y)}{\partial x_i} = \begin{cases} 0, \text{ if } i \neq \underset{i \in \{1,\cdots,n\}}{\operatorname{argmax}} \{q_i f_{x_i', y_i', r_i'}(x,y)\} \\ \frac{\alpha q_i h_{x_i', y_i', r_i'}(x,y)(x - x_i) \mathbb{1}_{[0,W]}(x_i)}{\sqrt{(x - x_i')^2 + (y - y_i')^2}}, \text{ o/w} \end{cases} \tag{4}$$

$$\frac{\partial \bar{\boldsymbol{M}}(x,y)}{\partial r_i} = \begin{cases} 0, \text{ if } i \neq \underset{i \in \{1,\cdots,n\}}{\operatorname{argmax}} \{q_i f_{x_i', y_i', r_i'}(x,y)\} \\ \alpha q_i h_{x_i', y_i', r_i'}(x,y) \mathbb{1}_{[R_{min}, R_{max}]}(r_i), \text{ o/w} \end{cases} \tag{5}$$

$$\frac{\partial \bar{\boldsymbol{M}}(x,y)}{\partial q_i} = \begin{cases} 0, \text{ if } i \neq \underset{i \in \{1,\cdots,n\}}{\operatorname{argmax}} \{q_i f_{x_i', y_i', r_i'}(x,y)\} \\ f_{x_i', y_i', r_i'}(x,y), \text{ o/w} \end{cases} \tag{6}$$

CircleOpt: Pixel-to-circle Optimization

- The loss function can be defined as:

$$L = L_{l2} + L_{pvb} + \gamma L_s, \tag{7}$$

where L_s is the sparsity regularize that can be formulated as:

$$L_s = \sum_{i=1}^{n} |q_i|. \tag{8}$$

Experiments

Comparison between CircleRule and SOTA pixel-level methods

Model	L2	PVB	EPE	#Shots
Develset[5]	39992.9	46251.7	8.2	699.8
CircleRule+Develset	49231.1	43407.1	14.4	123.8
NeuralILT[6]	37878.9	51092.9	8.1	332.1
CircleRule+NeuralILT	41720.6	50420.7	11.9	149.9
MultiILT[7]	27171.0	39854.0	2.9	271.5
CircleRule+MultiILT	35790.0	40725.0	8.3	260.0

Table: Comparison between CircleRule and SOTA pixel-based OPC methods. We show the averaged results for each metric on ICCAD 2013 benchmark.

[5] Guojin Chen et al. (2021). "DevelSet: Deep neural level set for instant mask optimization". In: *ICCAD*.

[6] Bentian Jiang et al. (2021). "Neural-ILT 2.0: Migrating ILT to Domain-Specific and Multitask-Enabled Neural Network". In: *TCAD*.

[7] Shuyuan Sun et al. (2023). "Efficient ILT via Multi-level Lithography Simulation". In: *DAC*.

Comparison between CircleRule and CircleOpt

Table: Mask Printability, Complexity Comparison for CircleRule and CircleOpt.

Bench	Area(nm^2)	CircleRule+Develset				CircleRule+NeuralILT				CircleRule+MultiILT				CircleOpt			
		L_2	PVB	EPE	#shots	L_2	PVB	EPE	#shots	L_2	PVB	EPE	#shots	L_2	PVB	EPE	#shots
case1	215344	59632	51087	23	102	53005	57775	13	170	49293	46945	8	253	43358	46905	3	214
case2	169280	53240	42993	19	80	45662	53770	13	128	39583	38565	2	236	35496	37920	1	215
case3	213504	107342	70906	69	205	93116	92536	61	213	105217	64381	69	242	75206	66241	32	194
case4	82560	21512	25623	7	44	18912	28873	6	73	14209	23649	2	164	13205	23234	1	104
case5	281958	53242	52706	4	140	52714	53575	8	192	32398	51711	0	317	34938	53110	1	172
case6	286234	52837	48595	5	203	48805	52200	7	190	38767	48290	1	313	36797	44629	0	252
case7	229149	36973	43124	0	146	24560	47353	0	146	17391	38744	0	335	21036	41118	0	175
case8	128544	18209	22917	1	65	16730	27114	2	89	12516	19968	0	181	13906	19859	0	169
case9	317581	62119	59295	8	214	53743	70986	9	221	40871	58311	1	407	47844	54625	1	251
case10	102400	27205	16825	8	39	9959	20025	0	77	9034	16694	0	152	9107	16969	0	83
Average		49231.1	43407.1	14.4	**123.8**	41720.6	50420.7	11.9	149.9	35790.0	40725.0	8.3	260	**33089.3**	**40451.5**	**3.9**	183

† *L_2 and PVB unit: nm^2.*

Ablation study: Sparsity Regularizer

Method	L2	PVB	EPE	#Shots
CircleOpt w/o Sparsity	**32595**	**40193**	4.1	208
CircleOpt	33089	40451	**3.9**	**183**

Table: Ablation study on the sparsity regularizer.

Q & A