

# A GPU-enabled Level Set Method for Mask Optimization

Ziyang Yu, Guojin Chen, Yuzhe Ma, Bei Yu  
 Department of Computer Science and Engineering  
 The Chinese University of Hong Kong

**Abstract**—As the feature size of advanced integrated circuits keeps shrinking, resolution enhancement technique (RET) is utilized to improve the printability in the lithography process. Optical proximity correction (OPC) is one of the most widely used RETs aiming at compensating the mask to generate a more precise wafer image. In this paper, we put forward a level-set based OPC with high mask optimization quality and fast convergence. In order to suppress the disturbance of the condition fluctuation in lithography process, we propose a new process window-aware cost function. Then, a novel momentum-based evolution technique is adopted, which demonstrates substantial improvement. Moreover, graphics processing unit (GPU) is leveraged for accelerating the proposed algorithm. Experimental results on ICCAD 2013 benchmarks show that our algorithm outperforms all previous OPC algorithms in terms of both solution quality and runtime overhead.

## I. INTRODUCTION

In the past decades, much progress has been made in optical lithography technology. In lithography process, pixelated optical masks are shaped in design patterns, and projected on the wafer images. However, the resolution of lithography system is proportional to the wavelength of the lithographic source light, and it's inversely proportional to the size of the mask due to the diffraction effect [1]. Thus, it becomes more and more challenging to further downscale the transistor since the feature size is already much smaller than the light source wavelength (193 nm).

To further extend the resolution limit, several resolution enhancement techniques (RETs) are proposed for mask optimization. Optical proximity correction (OPC) as a major RET, aims at compensating for the distortion of the printed image by pre-distorting the shape of the mask pattern. ILT as an important OPC method, treats the mask optimization as an inverse imaging problem which can be solved numerically. It aims at optimizing the carefully designed objective function and adjusting the pixel-wise mask backwards. A variety of attempts have been made in ILT to improve both the printed pattern fidelity and the process robustness [2]–[6].

However, the mask optimized by the pixel-wise ILT still contains unwanted tiny isolated stains and edge glitches, creating the obstacles to the mass production. As an alternate ILT strategy, level set algorithm has been widely explored [7]–[10]. Different from regarding every pixel on the mask as an isolated unit, level set method tracks the evolution of the mask boundary to reduce the geometric deviation in final printed image [11]. This improves the mask continuity and suppresses the degree of irregularity.

In this paper, we develop a comprehensive developed level set method to acquire better fidelity printed pattern. The major contributions of our work can be listed: (i) We propose a novel process variation based cost function, which could suppress the process variation band meanwhile reduce the edge placement error. (ii) We develop effective conjugate gradient method to improve the convergence. (iii) We adopt GPU acceleration scheme and reduce the time of the optimization notably. (iv) We perform experiments on ICCAD 2013 contest benchmarks and the results turn out to be prominent among the top winner on the contest and some previous algorithms.

The rest of the paper is organized as follows. Section II gives an introduction of the lithography process and a problem formulation. Section III gives the detailed elaboration of the optimization algorithm.

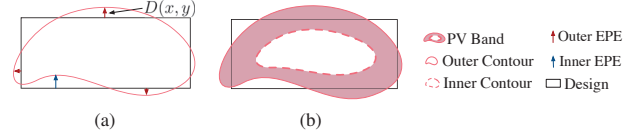


Fig. 1 Two considered metrics: (a) The measurement of edge placement error; (b) The measurement of PV Band.

Section IV details experimental results, followed by conclusion in Section V.

## II. PRELIMINARIES

In this section we introduce the preliminaries on the lithography model and the mask optimization problem.

### A. Lithography Process

The lithography process is composed of a projection optical model and a photo-resist model. The first model transfers the incident light containing information of mask pattern  $\mathbf{M}(x, y)$  into the aerial image  $\mathbf{I}(x, y)$  on the wafer plane. Due to the diffraction effect, the aerial image can be expressed:

$$\mathbf{I}(x, y) = \sum_{k=1}^K \mu_k |\mathbf{h}_k(x, y) \otimes \mathbf{M}(x, y)|^2. \quad (1)$$

In above equation,  $\mathbf{h}_k(x, y)$  is optical kernel functions. We apply  $K$ th order approximation to simplify the simulation,  $K = 24$  is the total number of optical kernels in accordance to the contest [12].

The aerial image  $\mathbf{I}$  is then transformed into the wafer image  $\mathbf{R}$  in the photo-resist process by comparing the aerial intensity to the photo-resist intensity threshold. To simulate this process, we adopt the constant threshold model here, the mathematical expression is given as:

$$\mathbf{R}(x, y) = \begin{cases} 1, & \text{if } \mathbf{I}(x, y) \geq I_{th}, \\ 0, & \text{if } \mathbf{I}(x, y) < I_{th}, \end{cases} \quad (2)$$

where  $I_{th}$  is the intensity threshold controls the binary image on the wafer plane.

### B. Inverse Lithography Technique

Due to the low-pass property of the band-limited lithography system, the printed wafer image  $\mathbf{R}$  is typically a blurred version of the input mask  $\mathbf{M}$  as is written in Equations (1) and (2). The objective is to synthesize a pre-distorted binary mask so that the corresponding printed image could be as close to the target image  $\mathbf{R}^*$  as possible. Mathematically, this equals to minimizing the geometric distance between the nominal printed image with the target image:

$$\mathbf{M}^* = \underset{\mathbf{M}}{\operatorname{argmin}} \|\mathbf{R} - \mathbf{R}^*\|^2. \quad (3)$$

In above equation,  $\|\cdot\|$  is the Euclidean norm ( $\mathcal{L}_2$  norm). Once the image is printed on the wafer plane, several metrics are measured to evaluate the quality of the image. In following parts, introduction of two metrics, called edge placement error and process variation band will be given.

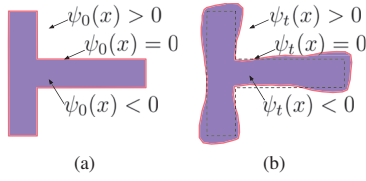


Fig. 2 Evolution process: (a) initial mask; (b) mask pattern after  $t$  iterations.

### C. Edge Placement Error

Edge placement error (EPE) is evaluated as the geometric distortion of the target image. As is shown in Fig. 1(a), probe points are set equidistantly on every horizontal and vertical edge. If the distance  $D$  from target to the printed image is larger than the EPE constraint  $th_{EPE}$ , we label it as an EPE violation.

$$\text{EPE violation}(x, y) = \begin{cases} 1, & \text{if } D(x, y) \geq th_{EPE}, \\ 0, & \text{if } D(x, y) < th_{EPE}. \end{cases} \quad (4)$$

### D. Process Variation Band

In real applications, process variation could cause the deviation in final printed image, leading to printed failure. It's essential to maintain the robustness of printed image. Process Variation Band (PV Band) is utilized to evaluate such robustness, which is defined as the XOR (exclusive or) region between the outermost and innermost contour under different process conditions, as is shown in Fig. 1(b). In this paper, the depth of the focus (focus/defocus) and intensity of incident light (dose) are considered as the process variables.

## III. ALGORITHMS

In level set framework, the mask optimization process is transformed into a contour evolution problem. Different from many ILT methods in which each pixel is taken as an optimized unit, level set method considers the mask boundary as a continuum. Fig. 2 gives an example of the boundary evolution process. The initial position for the mask boundary is represented by red contour in Fig. 2(a), the contour divides the mask plane into three parts which could be characterized by the level set function  $\psi(x, y)$ . The function values outside and inside the mask contour are set to be positive and negative respectively, and the contour itself is zero boundary. The mathematical representation is given as

$$\psi(x, y) = \begin{cases} -d(x, y), & \text{if } (x, y) \in C^-, \\ 0, & \text{if } (x, y) \in \partial C, \\ d(x, y), & \text{if } (x, y) \in C^+, \end{cases} \quad (5)$$

where  $d(x, y)$  is the Euclidean distance from  $(x, y)$  to the mask contour  $C$ .  $\partial C$  is the boundary of the mask.  $C^-$  and  $C^+$  are the inner and outer region of the mask respectively. Once the initial mask contour is given by  $\psi_0$ , the wafer image could be simulated and evaluated. After that error information could be sent back to update the level set function. The new mask is adjusted based on the changed function value. This is a iterative method and the mask could be updated to optimum, as is shown in Fig. 2.

### A. Level Set Based ILT

For level set based ILT, we adopt the binary mask. Whether the pixel is transmitted or blocked is determined by the level set function  $\psi(x, y)$ , the relation can be described:

$$\mathbf{M}(x, y) = \begin{cases} \mathbf{m}_{in}, & \text{if } \psi(x, y) \leq 0, \\ \mathbf{m}_{out}, & \text{if } \psi(x, y) > 0. \end{cases} \quad (6)$$

In above equation  $\mathbf{m}_{in} = 1$  and  $\mathbf{m}_{out} = 0$  mean that the pixel on  $(x, y)$  is determined to be inner and outer of the mask. In general,

this level set based ILT algorithm is developed to reduce the pattern distortion which is described in Equation (3). The cost function can be expressed as:

$$L_{nom}(\mathbf{M}) = \|\mathbf{R} - \mathbf{R}^*\|_F^2. \quad (7)$$

To enable the back propagation, we use *sigmoid* function to approximate the step function in Equation (2):

$$\mathbf{R} = \text{sig}(\mathbf{I}) = \frac{1}{1 + e^{-s(\mathbf{I} - I_{th})}}, \quad (8)$$

where  $s$  is the steepness. By combining Equations (1), (7) and (8), we could get the detailed expression of the nominal cost function:

$$L_{nom}(\mathbf{M}) = \left\| \text{sig} \left[ \sum_{k=1}^K \mu_k |\mathbf{h}_k(x, y) \otimes \mathbf{M}(x, y)|^2 \right] - \mathbf{R}^* \right\|_F^2. \quad (9)$$

Given the cost function, the velocity  $\mathbf{v}(x, y)$  can be deduced based on [13]:

$$\mathbf{v}(x, y) = -\frac{\partial L_{nom}(\mathbf{M})}{\partial \mathbf{M}} |\nabla \psi(x, y)|, \quad (10)$$

where  $|\nabla \psi(x, y)|$  is the gradient of level set function. We need to calculate the Jacobian of the cost function  $L_{nom}(\mathbf{M})$  at  $\mathbf{M}$ :

$$\begin{aligned} \frac{\partial L_{nom}(\mathbf{M})}{\partial \mathbf{M}} &= G(\mathbf{M}) = \frac{\partial \|\mathbf{R} - \mathbf{R}^*\|_F^2}{\partial \mathbf{M}} = 2(\mathbf{R} - \mathbf{R}^*) \frac{\partial \mathbf{R}}{\partial \mathbf{M}} \\ &= 2(\mathbf{R} - \mathbf{R}^*) \mathbf{R} (1 - \mathbf{R}) \frac{\partial}{\partial \mathbf{M}} \sum_{k=1}^K (\mu_k |\mathbf{h}_k(x, y) \otimes \mathbf{M}(x, y)|^2) \\ &= 2(\mathbf{R} - \mathbf{R}^*) \mathbf{R} (1 - \mathbf{R}) \\ &\quad \odot \frac{\partial}{\partial \mathbf{M}} \sum_{k=1}^K \mu_k (\mathbf{h}_k(x, y) \otimes \mathbf{M}(x, y)) \odot (\mathbf{h}_k^\dagger(x, y) \otimes \mathbf{M}(x, y)) \\ &= 2 \sum_{k=1}^K \mu_k \cdot \{ \mathbf{h}_k \otimes [(\mathbf{R}^* - \mathbf{R}) \odot \mathbf{R} \odot (1 - \mathbf{R}) \odot (\mathbf{h}_k^\dagger \otimes \mathbf{M})] \\ &\quad + \mathbf{h}_k^\dagger \otimes [(\mathbf{R}^* - \mathbf{R}) \odot \mathbf{R} \odot (1 - \mathbf{R}) \odot (\mathbf{h}_k \otimes \mathbf{M})] \}. \end{aligned} \quad (11)$$

### B. Process Variation Based Cost Function

To keep the optimization algorithm robust to different process conditions, a process variation band based cost function should be taken into consideration. We propose a new cost function which could help minimize the PV band area and meanwhile improve the quality of printed image without aggravating increasing computational burden:

$$L_{pvb}(\mathbf{M}) = \|\mathbf{R}_{in}(\mathbf{M}) - \mathbf{R}^*\|^2 + \|\mathbf{R}_{out}(\mathbf{M}) - \mathbf{R}^*\|^2. \quad (12)$$

To achieve a low cost value in Equation (12),  $\mathbf{R}_{in}(\mathbf{M})$  and  $\mathbf{R}_{out}(\mathbf{M})$  should be both close to the target image, which is stricter than minimizing the distance between  $\mathbf{R}_{in}(\mathbf{M})$  and  $\mathbf{R}_{out}(\mathbf{M})$ . This adjusted cost function could guide the optimization process to find desired mask, which could generated wafer image with tolerable PV band and less EPE number.

The total cost function is a linear combination of Equation (7) and Equation (12):

$$L = L_{nom}(\mathbf{M}) + w_{pvb} L_{pvb}(\mathbf{M}). \quad (13)$$

The gradient of total cost function can also be expressed in a separated way:

$$G(\mathbf{M}) = G_{nom}(\mathbf{M}) + w_{pvb} G_{pvb}(\mathbf{M}), \quad (14)$$

where  $w_{pvb}$  is the weight of the PV band cost function. The calculation of  $G_{pvb}(\mathbf{M})$  is similar to Equation (11).

### C. Conjugate Gradient Method

Conjugate gradient (CG) methods are proved to be efficient when solving optimization problems in large scale system. In our work, a CG method is applied to help achieve a better convergence. Denoting the velocity in  $i$ th iteration as  $\mathbf{v}_i$ :

$$\mathbf{v}_i = \begin{cases} -G(\mathbf{M})|\nabla\psi_0|, & \text{if } i = 0, \\ -G(\mathbf{M})|\nabla\psi_i| + \lambda_i\mathbf{v}_{i-1}, & \text{if } i > 0, \end{cases} \quad (15)$$

where  $\lambda_i$  is the conjugate gradient coefficient characterized by specific CG method. We adopt the Polak-Ribiere-Polyak (PRP) CG method proposed in [14]. In PRP method, the expression of CG coefficient is:

$$\lambda_i^{PRP} = \frac{\|G_i(\mathbf{M})|\nabla\psi_i|\|^2 - G_i(\mathbf{M})|\nabla\psi_i| \cdot G_{i-1}(\mathbf{M})|\nabla\psi_{i-1}|}{\|G_{i-1}(\mathbf{M})|\nabla\psi_{i-1}\|^2}. \quad (16)$$

By utilizing the gradient information in the former iteration, PRP algorithm could adjust the searching direction adaptively. The jamming could thus be effectively prevented.

### D. Level Set Based ILT Mask Update Algorithm

Based on the technics introduced above, we could build a complete iterative algorithm to generate the optimal mask using level set function. In initialization stage, the mask  $\mathbf{M}_0$  is initialized in the same shape with target  $\mathbf{R}^*$ . Then level set function  $\psi_0(x, y)$  for the initial mask can be calculated with Equation (5) (line 1). The mask is sent into forward lithography simulator and printed on wafer plane described by Equations (1) and (8). Once the wafer image  $\mathbf{R}_0$  is generated, gradient of total cost function  $G_0(\mathbf{M})$  is calculated by Equation (11) (line 2). From Equation (15), the starting velocity  $\mathbf{v}_0$  is set as the negative initial gradient (line 3).

In every iteration, the optimization process starts from choosing a proper time step  $\Delta t_i$  (line 5). In order to keeping the stability, we suppress the value of  $\Delta t_i$  with regarding to the maximum value of evolution velocity:  $\Delta t_i = \lambda_t / \max(|\mathbf{v}_i(x, y)|)$ . Then the change of level set function could be calculated (line 6). The level set function in the next iteration could be updated (line 7), leading to the correction of mask pattern (line 8). After that the wafer image is simulated and evaluated (line 9), the gradient for the next iteration could be computed (line 10), followed by the calculation of new velocity (line 12). The loop will continue until the pre-set iteration number  $N$  is achieved or the maximum velocity is less than the tolerance  $\epsilon$ , which means the optimization process has found a stable result. The whole process is described in Algorithm 1.

The updated mask in the last iteration loop is chosen as the optimized mask  $\mathbf{M}^*$ .

### E. GPU-enabled Acceleration

A GPU is formed by multiple units named streaming multiprocessors (SM). Each SM can execute many threads concurrently. Compute unified device architecture (CUDA) is a parallel computing platform and programming model developed by NVIDIA for its GPU. As is expressed in Equation (1), the level set method requires massive calls of forward lithography simulation which brings a large amount of computational efforts from convolution operations. Based on the properties of convolution, we can transform the calculation of aerial image intensity into the following expression:

$$\mathbf{M} \otimes \mathbf{H} = \sum_{k=1}^K \mu_k \cdot (\mathbf{M} \otimes \mathbf{h}_k) = \sum_{k=1}^K \mathbf{M} \otimes (\mu_k \cdot \mathbf{h}_k) = \mathbf{M} \otimes \sum_{k=1}^K \mu_k \cdot \mathbf{h}_k, \quad (17)$$

where  $\mathbf{H}$  is the general kernel function defined as the weighted sum of the optical kernel functions. With this transformation, the general

### Algorithm 1 Level-set based ILT method flow

**Require:** : Target image  $\mathbf{R}^*$ , optical kernels  $\mathbf{h}_1, \dots, \mathbf{h}_k$ , resistant model steepness  $s$ , intensity threshold  $I_{th}$ , max iteration number  $N$ , velocity tolerance  $\epsilon$ .

**Ensure:** Optimized mask  $\mathbf{M}^*$ .

- 1: Initialize:  $\psi_0 \leftarrow \mathbf{M}_0$ ;
- 2:  $G_0(\mathbf{M}) \leftarrow \frac{\partial L_0(\mathbf{M})}{\partial \mathbf{M}}$ ;
- 3:  $\mathbf{v}_0 = -G_0(\mathbf{M})|\nabla\psi(x, y)|$ ;
- 4: **repeat**
- 5: Time step:  $\Delta t_i \leftarrow \frac{\lambda_t}{\max(|\mathbf{v}_i(x, y)|)}$ ;
- 6: Level-set function:  $\psi_{i+1} \leftarrow \psi_i + \mathbf{v}_i(x, y)\Delta t_i$ ;
- 7: Mask pattern:  $\mathbf{M}_{i+1}(x, y) = \begin{cases} \mathbf{m}_{in}, & \text{if } \psi(x, y) \leq 0, \\ \mathbf{m}_{out}, & \text{if } \psi(x, y) > 0, \end{cases}$ ;
- 8: Evaluate the printed image with Score function:  $\mathbf{R}_{i+1}, (\mathbf{R}_{i+1})_{in}, (\mathbf{R}_{i+1})_{out}$ ;
- 9: Gradient of cost function:  $G_{i+1}(\mathbf{M}) \leftarrow \frac{\partial L_{i+1}(\mathbf{M})}{\partial \mathbf{M}}$ ;
- 10:  $\lambda_{i+1}^{PRP} \leftarrow \frac{\|G_i(\mathbf{M})|\nabla\psi_i|\|^2 - G_i(\mathbf{M})|\nabla\psi_i| \cdot G_{i-1}(\mathbf{M})|\nabla\psi_{i-1}|}{\|G_{i-1}(\mathbf{M})|\nabla\psi_{i-1}\|^2}$ ;
- 11:  $\mathbf{v}_{i+1} = -G_{i+1}(\mathbf{M})|\nabla\psi(x, y)| + \lambda_{i+1}^{PRP} \cdot \mathbf{v}_i$ ;
- 12: **until**  $i \geq N$  or  $|\mathbf{v}|_{max} \leq \epsilon$

kernel function could be precomputed in a multi-processing way. This could reduce convolution operations by  $K$  times and greatly improves the efficiency of our approach.

We apply fast Fourier transform (FFT) algorithm to accelerate convolution operations. By efficiently converting convolution operations between point-value representation and coefficient representation, FFT algorithm reduces the convolution computations from  $\mathcal{O}(N^2)$  to  $\mathcal{O}(N \log(N))$ . We implement FFT algorithm using CUDA toolkit for GPU-acceleration. Our GPU-based FFT algorithm provides a simple interface to compute FFTs by leveraging parallelism of the GPU, which reduces the total runtime to a large extend. Moreover, Equation (17) can be computed with GPUs in parallel to further reduce the total runtime.

## IV. EXPERIMENTAL RESULTS

Our level set ILT algorithm is implemented in C/C++ on an Intel Xeon E5-2690 V4 CPU with 2.6 GHz and 32 GB RAM. The GPU acceleration experiments are tested on a single Nvidia Tesla P100 GPU accelerator (16GB card). We adopt the 193nm wavelength lithography system with a defocus range of  $\pm 25$  nm and a dose range of  $\pm 2\%$ , which is provided in the ICCAD 2013 contest [12]. We adopt 24 optical kernel functions in optical model, the threshold intensity in photoresist model is 0.225. The PV band metric system and EPE checker modules are also provided in the contest. Ten benchmarks composed of rectangles and polygons in different sizes released by IBM [12] are tested. Each benchmark layout is a 32nm 1x Metal layer. The size of the image is 2048nm  $\times$  2048nm with the resolution of 1nm<sup>2</sup> per pixel. The outermost final printed pattern is generated at nominal focus and +2% dose while the innermost printed pattern is generated at defocus and -2% dose. The EPE violation threshold  $th_{EPE}$  is set to 15nm. EPE is measured on the sample points locate on the pattern edges every 40nm.

To evaluate the effectiveness of our method quantitatively, we adopt four metrics described in [12], which are number of edge placement error (#EPE), process variation band area (PVB), runtime(RT) and number of shape violations. The score function is the linear combination of those metrics:

$$\text{Score} = \text{RT} + 4 \times \text{PVBand} + 5000 \times \text{\#EPE} + 10000 \times \text{ShapeViol}. \quad (18)$$

In above score function, ShapeViol is the number of shape violations visually checked from the final printed image.

We compare our results introduced before with the four recent

TABLE I Comparison with top winners of ICCAD 2013 contest and previous algorithms

Test case ID	Pattern area	MOSAIC_fast [6]			MOSAIC_exact [6]			robust OPC [15]			PVOPC [16]			Ours		
		#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score	#EPE	PVB	Score
B1	215344	6	58232	263246	9	56890	274267	0	66218	265150	2	58269	243240	4	62693	270895
B2	169280	10	47139	238812	4	48312	214493	0	53434	213878	0	52674	210826	1	50724	207977
B3	213504	59	82195	624101	52	84608	600955	18	146776	677256	47	81541	561367	29	100945	598994
B4	82560	1	28244	118298	3	24723	115161	0	33266	133371	0	26960	108030	0	29831	119508
B5	281958	6	56253	255327	2	56299	237363	1	65631	267713	4	61820	267342	1	56510	231116
B6	286234	1	50981	209238	1	49285	204224	0	62068	248625	0	55090	220414	1	51204	209881
B7	229149	0	46309	185475	0	46280	186761	0	51069	204495	0	51977	207982	0	45056	180288
B8	128544	2	22482	100186	2	22342	100031	0	25898	103691	0	22869	91541	1	22757	96095
B9	317581	6	65331	291646	3	62529	268138	1	75387	306667	0	70713	282907	0	64597	258466
B10	102400	0	18868	75703	0	18141	73276	0	18536	74205	0	17846	71425	0	18769	75140
Avg.				236203			227467			249505			226507			<b>224836</b>

Pattern area / PVB unit:  $nm^2$ 

TABLE II Runtime comparison

Case	MOSAIC [6]		robust OPC [15]	PVOPC [16]	Ours	
	Fast	Exact			CPU	GPU
B1	318	1707	278	164	365	123
B2	256	1245	142	130	303	81
B3	321	2523	152	203	902	214
B4	322	1269	307	190	591	184
B5	315	2167	189	62	218	76
B6	314	2084	353	54	223	65
B7	239	1641	219	74	220	64
B8	258	663	99	65	200	67
B9	322	3022	119	55	219	63
B10	231	712	61	41	206	64
Avg.	289.6	1703.3	191.9	103.8	344.7	<b>100.1</b>

Runtime unit: *second*

process window-aware pixel based OPC methods: MOSAIC\_fast, MOSAIC\_exact in [6], robust OPC in [15] and PVOPC in [16]. The results and score comparisons (ratio) are recorded in Table I. It's worth mentioning the authors of [15] offered their detailed data to us, based on which we are able to compute the score values using Equation (18). Although the tradeoff between EPE counts and PV band area is still challenging to handle, our method successfully attains the best general performance with lowest score among all the compared methods. The results show our method could generate robust and high fidelity pattern images on these benchmarks, and it reveals potential for achieving high quality masks for other desired patterns.

We display our runtime results tested on CPU and GPU respectively and compare them with the methods listed in TABLE I. The runtime is measured from the input of target pattern to the production of optimized mask. Results are shown in TABLE II. MOSAIC\_fast adopts an alternate gradient method to reduce computational time, the runtime is 16% better than our CPU results. Considering this algorithm is tested on a different machine with higher CPU frequency (3.4 GHz) and 32 GB memory, this difference is tolerable. MOSAIC\_exact sacrifices the computational efficiency to pursue high quality mask, we achieve  $4.94\times$  speedup on this. The robust OPC consumes 44% less time than ours. However, the simulations consume most time in the optimization process. As is explained in [15], they only run the simulators in two process conditions for each iteration and estimate the results in third process condition using the experiment data. For PVOPC, our run time on CPU is almost 70% longer.

By adopting GPU-acceleration we greatly reduce the computational time in all cases without degenerating the fidelity or robustness of the masks. It has 71% runtime reduction compared with our CPU result. We could achieve the fastest among all the listed methods. This runtime superiority could be essential to develop the industrial production efficiency.

## V. CONCLUSION

In this paper, we develop a level set inverse lithography mask optimization method with CUDA speedup to generate mask pattern with high fidelity and robustness in very short time. We formulate the new process variation based cost function to minimize the PV band and pattern displacement effectively. Conjugate gradient algorithm is adopted to help improve the convergence. We also use GPU to accelerate the optimization process. Numerical experimental results show that our method could produce better masks in short time, and it is robust to different process variations.

## ACKNOWLEDGMENT

This work is partially supported by The Research Grants Council of Hong Kong SAR (No. CUHK14209420).

## REFERENCES

- [1] L. Rayleigh, "Xxxi. investigations in optics, with special reference to the spectroscope," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1879.
- [2] A. Poonawala and P. Milanfar, "Mask design for optical microlithography—an inverse imaging problem," *IEEE TIP*, 2007.
- [3] —, "A pixel-based regularization approach to inverse lithography," *Microelectronic Engineering*, 2007.
- [4] N. Jia, A. K. Wong, and E. Y. Lam, "Robust mask design with defocus variation using inverse synthesis," in *Lithography Asia*, vol. 7140, 2008.
- [5] F. Liu and X. Shi, "An efficient mask optimization method based on homotopy continuation technique," in *Proc. DATE*, 2011.
- [6] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "Mosaic: Mask optimizing solution with process window aware inverse correction," in *Proc. DAC*, 2014.
- [7] Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," *Optics Express*, 2009.
- [8] Y. Shen, N. Jia, N. Wong, and E. Y. Lam, "Robust level-set-based inverse lithography," *Optics Express*, 2011.
- [9] W. Lv, S. Liu, Q. Xia, X. Wu, Y. Shen, and E. Y. Lam, "Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step," *Journal of Vacuum Science & Technology B*, 2013.
- [10] Z. Geng, Z. Shi, X.-L. Yan, K.-S. Luo, and W.-W. Pan, "Fast level-set-based inverse lithography algorithm for process robustness improvement and its application," *Journal of Computer Science and Technology*, 2015.
- [11] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, 1988.
- [12] S. Banerjee, Z. Li, and S. R. Nassif, "Iccad-2013 cad contest in mask optimization and benchmark suite," in *Proc. ICCAD*, 2013.
- [13] F. Santosa, "A level-set approach for inverse problems involving obstacles," *ESAIM: COCV*, 1996.
- [14] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, 1969.
- [15] J. Kuang, W.-K. Chow, and E. F. Young, "A robust approach for process variation aware mask optimization," in *Proc. DATE*, 2015.
- [16] Y.-H. Su, Y.-C. Huang, L.-C. Tsai, Y.-W. Chang, and S. Banerjee, "Fast lithographic mask optimization considering process variation," *IEEE TCAD*, 2016.