Cascadic multigrid algorithm for robust inverse mask synthesis in optical lithography

Wen Lv
Edmund Y. Lam
Haiqing Wei
Shiyuan Liu

Cascadic multigrid algorithm for robust inverse mask synthesis in optical lithography

Wen Lv,[a,b] **Edmund Y. Lam,**[b] **Haiqing Wei,**[a] **and Shiyuan Liu**[a,c,*]
[a]Huazhong University of Science and Technology, State Key Laboratory of Digital Manufacturing Equipment and Technology, Wuhan 430074, China
[b]The University of Hong Kong, Department of Electrical and Electronic Engineering, Pokfulam, Hong Kong
[c]Huazhong University of Science and Technology, Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China

Abstract. Robust inverse mask synthesis is computationally intensive, and its turnaround time continues to rise hand-in-hand with the ever-shrinking integrated circuit feature size. We report the development of a cascadic multigrid (CMG) algorithm for robust inverse mask synthesis, which starts from a relatively coarse mask grid and refines it iteratively in stages, so as to achieve significant speedup without compromising numerical accuracy. Since the CMG algorithm entails frequent changes of the computational grid size, we need to intentionally introduce an analytical circle-sampling technique for modeling the forward lithography imaging and employ an edge distance error as metric to guide mask synthesis. These two techniques work nicely with variable grid sizes and are well suited for our CMG algorithm. As a result, our algorithm achieves more than four times speedup over conventional methods that synthesize a mask on a fixed fine grid. Numerical results are presented to demonstrate the validity and efficiency of the proposed method. © *The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI.* [DOI: 10.1117/1.JMM.13.2.023003]

1 Introduction

In attempts to push the limit of optical lithography, several resolution enhancement techniques (RETs) have been widely applied to improve the lithographic performance.[1–3] Inverse lithography technology (ILT),[4,5] as an active approach of RETs, is considered as an effective and economically viable way to meet various challenges in current and future technology nodes.[6,7] To put ILT into practice, various methods have been proposed by academic as well as industrial communities on mask manufacturing rule constraints,[8,9] pattern grouping strategy to accelerate computing,[10] GPU-based hardware accelerated techniques,[11] and mathematical solution methods for the inverse problem.[4,5] In this paper, we focus on improvement of mathematical solution methods, where the computational efficiency is one of the most noteworthy issues. Currently, as critical dimension (CD) shrinks, the pattern density of integrated circuits gets much denser and lithographic process variations, such as lens-wafer defocus and exposure dose variation, become more pronounced. It, therefore, requires finer mask models and through process window compensated approaches, which in return significantly increase the computational burden.

Inverse lithography technique treats the mask design as an inverse mathematical problem that aims at synthesizing an input mask to deliver a desired output pattern on the wafer. Liu and Zakhor pioneered a mask design method based on the branch and bound algorithm and simulated annealing.[12,13] But this is a time-consuming process. In order to reduce the computational complexity, iterative methods were proposed to solve the inverse problem via an optimization process,[14,15] and they were further classified as linear, quadratic, and nonlinear optimization problems.[16,17] Meanwhile, Poonawala and Milanfar designed the model-based optical proximity correction system and introduced the steepest descent algorithm for the optimization framework.[18,19] Subsequently, the optimization framework was further generalized for phase-shifting masks[20,21] and partially coherent imaging systems,[22–26] and the optimization algorithm was improved with an active set method[21] and with an augmented Lagrangian method.[27] Most recently, Lv et al. further improved the computational efficiency of the mask synthesis by using the conjugate gradient and an optimal iterative step,[28] and by introducing a mask filtering technique to enhance the regularity of the synthesized mask pattern.[29,30] On the other hand, with ever-shrinking feature size, the printed feature becomes increasingly sensitive to the fluctuation of the fabrication process, which limits the yield in semiconductor industry. In order to synthesize masks that are robust to process variations, the average wafer performance is optimized via minimizing the expectation of the difference between the desired pattern and the output pattern with respect to process fluctuations.[31–36] This approach takes into account process variations explicitly, and is well understood and easily accomplished. However, it further increases the computational complexity. It is noted that all these methods discretize mask as a raster image constituted by grids (pixels) and perform mask synthesis on a certain fine grid throughout the mask synthesizing process. Since the computational complexity of such iterative methods is $O[K \times \log(K)]$,[14,17,18] where K is total grid numbers used in the discretization of mask, the mask grid size strongly impacts the computational cost. Naturally, one would like to employ a coarser grid to model mask

*Address all correspondence to: Shiyuan Liu, E-mail: shyliu@mail.hust.edu.cn

and, thus, to cut down the runtime. Unfortunately, such straightforward strategy usually leads to inaccurate simulation results and a poorly performing mask that fails to optimally utilize the imaging capability of the lithography scanner. There has been a tradeoff between the runtime and mask quality.

In this paper, a cascadic multigrid (CMG) algorithm[37,38] is introduced to improve the computational efficiency of robust inverse mask synthesis, while maintaining the model accuracy and mask pattern quality. The fundamental idea of CMG algorithms is to perform more iterations on coarser grids and fewer iterations on fine ones, so that the overall runtime may be dramatically reduced. The CMG algorithm is a one-way multigrid method that requires no coarse grid correction and is, thus, easy to implement. Since the computational complexity of iterative methods is $O[K \times \log(K)]$, the CMG algorithm significantly reduces the runtime compared to the conventional methods, which accomplish mask synthesis on a fixed fine grid. More details of the CMG algorithm will be presented in Sec. 3.3. For theoretical convergence results and complexity analysis on CMG algorithms, interested readers may refer to Refs. 37 to 40.

Moreover, we have recently developed an analytical circle-sampling technique for fast imaging simulation of partially coherent systems by decomposing the double-impulse response (DIR) function into analytical kernels.[41] The DIR function,[42] whose Fourier transform is the familiar transmission cross-coefficient (TCC), is projected onto the circle-sampling function (CSF)[41] space and converted into a much smaller projected matrix. Singular value decomposition (SVD) is performed to the smaller projected matrix, and then we obtain analytical optical kernels with its eigenvectors and the CSFs. We have demonstrated that this method avoids directly performing SVD to the large TCC matrix; the simulation speed is, thus, dramatically improved. In particular, the derived optical kernels have analytical forms; as a result, the grid size of the kernels can be set to any desired value without re-preforming decomposition. Also, most recently we developed a metric called edge distance error (EDE) to guide mask synthesis.[29,30] Compared to the commonly used metric pattern error,[16–28] the metric EDE has a dimension of length and is independent of the simulation grid size. The analytical circle-sampling technique and EDE are both independent of grid size. Such nature is well suited for the CMG algorithm, which frequently changes the mask grid size during its synthesizing process. In this paper, we will apply the analytical circle-sampling technique to the forward lithography imaging modeling and employ EDE to guide mask synthesis.

The remainder of this paper is organized as follows. Section 2 details the forward lithography imaging model using an analytical circle-sampling technique. Section 3 introduces the robust inverse lithography problem and presents CMG algorithm in detail. Section 4 provides simulation results to demonstrate the validity and efficiency of the proposed method. Finally, we draw some conclusions in Sec. 5.

2 Forward Lithography Imaging Model

We apply an analytical circle-sampling technique to the forward lithography imaging simulation. The lithography imaging process is often decomposed into two parts, namely

the optical image formation and the resist development. The optical image $I(\mathbf{r})$ generated by a partially coherent imaging system can be expressed by a bilinear transform in the spatial domain as[43]

$$I(\mathbf{r}) = \int \int D(\mathbf{r} - \mathbf{r}_1, \mathbf{r} - \mathbf{r}_2)M(\mathbf{r}_1)M^\dagger(\mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2, \quad (1)$$

where $\mathbf{r}$ represents the spatial coordinate (x, y), $M(\mathbf{r})$ is the mask transmittance function, $\dagger$ denotes complex conjugation, and

$$D(\mathbf{r}_1, \mathbf{r}_2) = J(\mathbf{r}_1 - \mathbf{r}_2)H(\mathbf{r}_1)H^\dagger(\mathbf{r}_2) \quad (2)$$

is called the DIR function whose Fourier transform is the familiar TCC.[42] Here, $J(\mathbf{r}_1, \mathbf{r}_2)$ is the mutual intensity function that describes the coherence of the illumination source, and $H(\mathbf{r})$ is the point spread function (PSF) of the optical system that may be obtained by an inverse Fourier transform of the pupil function $\tilde{H}(\mathbf{f})$ as

$$H(\mathbf{r}) = \mathrm{IFT}\{\tilde{H}(\mathbf{f})\}, \quad (3)$$

where $\tilde{H}(\mathbf{f})$ is the pupil function without any aberrations and $\mathbf{f}$ is the normalized pupil plane coordinate.

Since the DIR function is Hermitian and band-limited, it is possible to approximate the DIR function as a finite CSF series as[41]

$$D(\mathbf{r}_1, \mathbf{r}_2) = \sum_{k=1}^{N} \sum_{l=1}^{N} p_{k,l}\varphi_k(\mathbf{r}_1)\varphi_l^\dagger(\mathbf{r}_2), \quad (4)$$

where N is the total number of CSFs, $\varphi_k(\mathbf{r})$ is the k'th CSF, and $p_{k,l}$ are the expansion coefficients. It is noted that the projected matrix $\mathbf{P}$, i.e., $[p_{k,l}]$, is also Hermitian and positive definite, so that we can perform SVD to further decompose the projected matrix $\mathbf{P}$ into eigenvalues and eigenvectors, as follows:

$$\mathbf{P} = \sum_{i=1}^{Q} \mu_i \mathbf{s}_i \mathbf{s}_i^H, \quad (5)$$

where $\mathbf{s}_i$ is the i'th eigenvector with Q eigenvectors in total, $\mathbf{s}_i^H$ is its Hermitian conjugate, and μ_i is the corresponding eigenvalue. Substituting Eq. (5) into Eq. (4), we can analytically express the DIR function as
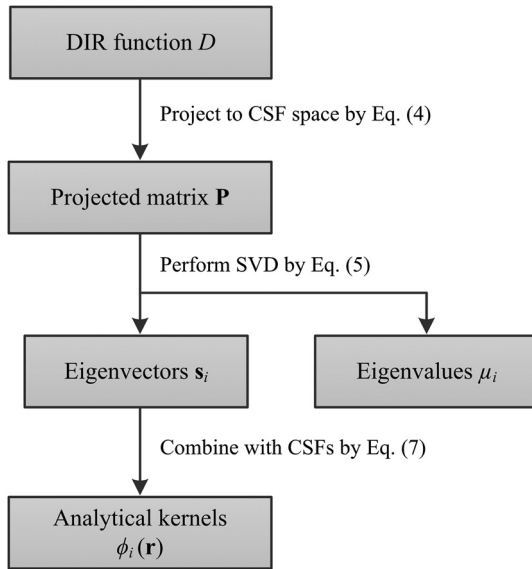
$$D(\mathbf{r}_1, \mathbf{r}_2) = \sum_{i=1}^{Q} \mu_i \phi_i(\mathbf{r}_1)\phi_i^\dagger(\mathbf{r}_2), \quad (6)$$

where $\phi_i(\mathbf{r})$ is called the i'th analytical optical kernel, which is calculated as

$$\phi_i(\mathbf{r}) = \sum_{l=1}^{N} \mathbf{s}_{i,l}\varphi_l(\mathbf{r}), \quad (7)$$

where $\mathbf{s}_{i,l}$ is the l'th element of $\mathbf{s}_i$. Figure 1 illustrates the flow chart of the DIR function decomposition into analytical kernels.

Substituting Eq. (6) into Eq. (1), the optical image is given by

Fig. 1 Flow chart of the double-impulse response function decomposition into analytical kernels.

Fig. 2 Pixel-based representation of a mask pattern and its output pattern on the wafer.

$$I(\mathbf{r}) = \sum_{i=1}^{Q} \mu_i |\phi_i(\mathbf{r}) \otimes M(\mathbf{r})|^2. \qquad (8)$$

Subsequently, the optical image $I(\mathbf{r})$ goes through the resist development to form the printed image on the wafer. The resist effect is often approximated by a constant threshold resist model using the following logarithmic Sigmoid function:[18]

$$\text{sig}[I(\mathbf{r})] = \frac{1}{1 + \exp\{-a[I(\mathbf{r}) - t]\}}, \qquad (9)$$

where a is the steepness of the Sigmoid function and t is the threshold level of the resist.

Putting Eqs. (8) and (9) together, the lithography imaging equation can be formulated as

$$\begin{aligned} Z(\mathbf{r}) = \Gamma\{M(\mathbf{r})\} &= \text{sig}[I(\mathbf{r})] \\ &= \text{sig}\left[\sum_{i=1}^{Q} \mu_i |\phi_i(\mathbf{r}) \otimes M(\mathbf{r})|^2\right], \end{aligned} \qquad (10)$$

where the operator $\Gamma\{\cdot\}$ implements the forward mapping from the input mask $M(\mathbf{r})$ to the output pattern $Z(\mathbf{r})$ on the wafer.

3 Inverse Mask Synthesis

Ideally, the output pattern on the wafer is desired to match the input design intent that serves as the start point in most mask synthesis routines. However, the optical imaging system typically acts as a low-pass spatial frequency filter and cannot deliver the high-frequency components of the intended pattern on wafer.[7] The band-limited system causes the output pattern to be a distorted version of the input intent. So, the objective of inverse lithography is to synthesize a mask pattern to precompensate the effect of frequency loss so as to deliver a wafer image closer to the desired pattern.
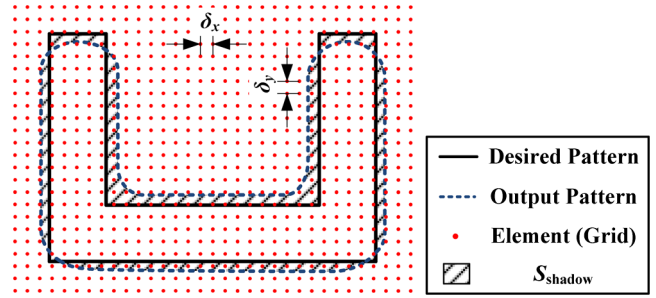
3.1 Evaluation Metrics

In order to evaluate the performance of the synthesized mask pattern, we introduce some evaluation metrics.

An EDE is introduced as a metric to evaluate the difference between the output pattern of the input mask and the desired pattern. Figure 2 depicts the pixel-based representation of a mask pattern and its output pattern on the wafer, where the red dots are discrete sampling elements (grids or pixels) of the patterns, S_{shadow} denotes the absolute difference area between the desired pattern contour and the output pattern contour, L is the perimeter of the desired pattern contour, and δ_x and δ_y are the lengths of the element along the x and y directions, respectively. EDE is defined as

$$\text{EDE} = \frac{S_{\text{shadow}}}{L}. \qquad (11)$$

Assuming that the element size is small enough in Fig. 2, the absolute difference area S_{shadow} can be approximated by multiplying the total number of elements N_s in shadow and the element area as

$$S_{\text{shadow}} = N_s \cdot (\delta_x \cdot \delta_y). \qquad (12)$$

Noticing that the value of the grid in the output pattern is either 0 or 1, the number N_s is approximately equal to the pattern error as

$$N_s = \|\Gamma\{M\} - Z^*\|_2^2, \qquad (13)$$

where $\|\cdot\|_2$ is the L_2 norm, Z^* is the desired pattern, and $\|\Gamma\{M\} - Z^*\|_2^2$ is usually called pattern error, which is often employed as a metric to evaluate the difference between the output pattern of the input mask and the desired pattern. However, pattern error is a dimensionless quantity and highly depends on mask feature and simulation grid size. Therefore, pattern error is inappropriate to act as a metric, especially when the simulation grid size is constantly changed. Substituting Eq. (13) into Eq. (12) and then into Eq. (11), we have the expression of EDE as

$$\text{EDE}(M) = \frac{\delta_x \cdot \delta_y}{L} \|\Gamma\{M\} - Z^*\|_2^2. \qquad (14)$$

It is noted that EDE has the dimension of length and, thus, is independent of the simulation grid size. More detailed physical description of EDE can be found in Ref. 30.

In addition, in order to quantify the manufacturability of the synthesized mask, mask quadratic error metric $R_Q(M)$

and mask complexity metric $R_{\text{TV}}(M)$ are adopted.[18,19] Since we focus on the binary mask in this paper, the mask quadratic error metric $R_Q(M)$ and the mask complexity metric $R_{\text{TV}}(M)$ are expressed, respectively, as

$$R_Q(\text{M}) = \int_\Omega [1 - (2M - 1)^2]\mathrm{d}\mathbf{r}, \tag{15}$$

$$R_{\text{TV}}(\text{M}) = \int_\Omega (|\nabla M_x| + |\nabla M_y|)\mathrm{d}\mathbf{r}, \tag{16}$$

where $\nabla M_x = \partial M/\partial x$, $\nabla M_y = \partial M/\partial y$, and Ω is the simulation area, or the total number of grids in mask M.

3.2 Formulation of Robust Inverse Lithography

From Sec. 3.1, there are, overall, three evaluation metrics that need to be optimized, i.e., the EDE, the mask quadratic error metric, and the mask complexity metric. In the literature, they are usually combined with certain proportions β_1 and β_2 to be stated as one single optimization objective as

$$F(M) = \text{EDE}(M) + \beta_1 R_Q(M) + \beta_2 R_{\text{TV}}(M). \tag{17}$$

Here, $F\{M\}$ is also called cost function. Therefore, the inverse lithography problem under the nominal condition is defined as

$$M^* = \arg \min_M F(M). \tag{18}$$

However, the formulation Eq. (18) of inverse lithography problem is under the assumption of an ideal imaging system without any process variations. However, in practice, process variations are significant and should not be ignored. In this paper, we take into account the two most important variations in lithography process, i.e., lens-wafer defocus and exposure dose variation, and employ a statistical strategy to optimize the average wafer performance via minimizing the expectation of the difference between the desired pattern and the output pattern with respect to process fluctuations. Specifically, exposure dose variation can be accounted for by varying the resist threshold t in Eq. (9); the effect of defocus h on the image intensity distribution is effectively an even-type lens aberration, leading to a defocused PSF function $H(\mathbf{r}, h)$.[32]

$$H(\mathbf{r}; h) = \text{IFT}\left[\tilde{H}(\mathbf{f}) \cdot \exp\left(j\pi h \frac{\text{NA}^2}{\lambda}\mathbf{f}^2\right)\right], \tag{19}$$

where $j = \sqrt{-1}$, NA is the numerical aperture, and λ is the incident light wavelength. Therefore, the expectation of the difference between the desired pattern and the output pattern with respect to process fluctuations is expressed by

$$\text{EDE}_s(M) = \sum_m^{N_h} \sum_n^{N_t} [\alpha(h_m) \cdot \zeta(t_n) \cdot \text{EDE}(M; h_m, t_n)], \tag{20}$$

where h_m are the discretized values of defocus h with a density distribution $\alpha(h_m)$; t_n are the discretized values of equivalent exposure dose t with a density distribution $\zeta(t_n)$, and N_h and N_t are the corresponding discretization number. $\text{EDE}_s(M)$ is called statistical EDE, and it takes process variations into account explicitly. In this paper, the statistical EDE is used to evaluate the average wafer performance with respect to process fluctuations, namely, to evaluate the robustness of mask.

Now, the cost function in the robust inverse lithography is expressed as

$$G(M) = \text{EDE}_s(M) + \beta_1 R_Q(M) + \beta_2 R_{\text{TV}}(M). \tag{21}$$

The robust inverse lithography problem is, thus, defined as

$$M^* = \arg \min_M G(M). \tag{22}$$

3.3 Cascadic Multigrid Algorithm

In this section, we develop a CMG algorithm to solve the robust inverse lithography problem. For the setup of CMG algorithm, we define a sequence of spaces

$$\{\Omega_1, \Omega_2, \ldots, \Omega_C\}, \tag{23}$$

with the corresponding space grid size

$$\delta_1 > \delta_2 > \ldots > \delta_C. \tag{24}$$

Since the grid is usually square, and we use only a parameter δ to describe the space grid size for simplicity, that means $\delta = \delta_x = \delta_y$. Ω_C denotes the finest grid space currently used for discretization of a mask. First, we solve the robust inverse lithography problem, i.e., Eq. (22), on the coarsest space Ω_1 and obtain a mask M_1^*. The obtained mask M_1^* on the space Ω_1 is then embedded in the next space Ω_2 via an interpolation operation, and then the interpolated mask M_2^0 servers as the initial value on the space Ω_2. With the initial value M_2^0, it calculates a mask M_2^* on the space Ω_2. The computation repeats in this manner until a mask M_C^* on the finest space Ω_C is obtained. The pseudo-code of CMG algorithm is described in Table 1.

The CMG algorithm starts mask synthesis on the coarsest space Ω_1. Since the coarsest mask contains much less optimization variables (grids) than the finest mask on space Ω_C, it spends much less runtime to find a solution and is more likely to change each variable. From the perspective of signal processing, the coarse mask can be considered as the low-frequency portions of mask. The low-frequency portions are first optimized and then further corrected by adding high-frequency details on the refined spaces. By contrast, conventional methods perform mask synthesis directly on the finest space Ω_C, where it is expensive to change each variable and impossible to separate and treat efficiently the low-frequency information content of the mask. Therefore, the CMG algorithm can achieve a mask pattern by taking less runtime with better solution performance, compared to the conventional methods. This is the fundamental rationale behind the CMG algorithm, and simulation results below demonstrate its advantages.

The procedure:

Setup: Set C spaces $\{\Omega_1, \Omega_2, \ldots, \Omega_C\}$, and initial mask
$M_1^0 = Z^*, M_1^0 \in \Omega_1$.

For $k = 1, 2, \ldots, C$ **do**

   1: Solve $M_k^* = \arg\min_M G(M), M \in \Omega_k$ with an initial value M_k^0;

   2: Prolong to next space Ω_{k+1}: $M_k^* \to M_{k+1}^0$;

   3: $k + 1 \to k$.

End For

Output: M_C^* is the optimal mask.

3.4 Conjugate Gradient Method

In this section, we present a solution procedure of solving Eq. (22) on a certain space with a conjugate gradient (CG) method. First of all, we calculate the gradient of the cost function $G(M)$ with respect to mask M as

$$\nabla G(M) = \nabla \text{EDE}_s(M) + \beta_1 \nabla R_Q(M) + \beta_2 \nabla R_{\text{TV}}(M). \tag{25}$$

Detailed expression of Eq. (25) is given in the Appendix. In CG method, the optimization direction v^k in the k'th iteration is defined by

$$v^k = \begin{cases} -g^k + \eta^k v^{k-1} & \text{if } k \geq 1 \\ -g^k & \text{if } k = 0 \end{cases}. \tag{26}$$

Here, g^k is a matrix representing the gradient $\nabla G(M)$ of cost function $G(M)$ at mask M, and η^k is a factor that depends on different CG methods. When $\eta^k = 0$, the evolution velocity is the negative gradient, and the CG method reduces to the steepest descent method. In this paper, we employ the Polak-Ribière-Polyak (PRP) CG method.[28] In practical computations, the PRP CG method is generally believed to be one of the most efficient CG methods. It makes use of the velocity information in the previous iteration and can automatically adjust η^k to avoid jamming. It essentially performs a restart when a bad direction occurs. In PRP CG method, η^k is defined as

$$\eta^k = \frac{\|g^k\|_2^2 - \sum g^k \cdot g^{k-1}}{\|g^{k-1}\|_2^2}. \tag{27}$$

Now, the optimization procedure is described as follows:

Iteration 0: Since the mask value is bound-constrained to [0, 1], we use the following parametric transformation:

$$M = \frac{1 + \cos(\theta)}{2}, \theta \in (-\infty, +\infty). \tag{28}$$

Then given a desired output pattern $Z^*(\mathbf{r})$, we compute the initial value θ^0.

$$M^0(\mathbf{r}) = \kappa_1 \cdot Z^*(\mathbf{r}) + \kappa_2, \tag{29}$$

$$\theta^0(\mathbf{r}) = \cos^{-1}(2M^0 - 1), \tag{30}$$

where κ_1 and κ_2 are parameters to adjust the initial mask value; for example, $\kappa_1 = 0.90$ and $\kappa_2 = 0.05$ in this paper. We do that because $M(i, j) = 0$ or 1 would degrade the gradient of location (i, j) to 0, and therefore, the optimization freedom would be reduced.[25,30] Finally, we calculate the initial gradient g^0 and the optimization direction v^0 based on Eq. (26).

$$g^0 = \nabla_{\theta^0} G(M) = \frac{\partial G}{\partial M^0} \frac{\partial M^0}{\partial \theta^0}, \tag{31}$$

$$v^0 = -g^0, \tag{32}$$

where

$$\frac{\partial M}{\partial \theta} = \frac{-\sin(\theta)}{2}. \tag{33}$$

Iteration k:

   Step i: Search the step length γ^k in the direction v^k,

$$\gamma^k = \arg \min_\gamma [G(\theta^k + \gamma \cdot v^k)]. \tag{34}$$

   Step ii: Update θ^{k+1},

$$\theta^{k+1} = \theta^k + \gamma^k \cdot v^k. \tag{35}$$

   Step iii: Calculate the optimization direction for the next iteration.

$$g^{k+1} = \nabla_{\theta^{k+1}} G(M) = \frac{\partial G}{\partial M^{k+1}} \frac{\partial M^{k+1}}{\partial \theta^{k+1}}, \tag{36}$$

$$\eta^{k+1} = \frac{\|g^{k+1}\|_2^2 - \sum g^{k+1} \cdot g^k}{\|g^k\|_2^2}, \tag{37}$$

$$v^{k+1} = -g^{k+1} + \eta^{k+1} \cdot v^k. \tag{38}$$

   If $\|v^{k+1}\| < \Lambda$ or $k > \Psi$, go to **Stop**.
   Else, return to **Step i**.

   Stop: Obtain the mask.

$$M^*(\mathbf{r}) = \frac{1 + \cos(\theta^{k+1})}{2}. \tag{39}$$

In the above procedure, the iteration is terminated when $\|v^{k+1}\| < \Lambda$ or $k > \Psi$, where Λ is defined as the minimum value of the norm of velocity and Ψ is the prescribed upper limit of the number of iterations. The termination criterion $\|v^{k+1}\| < \Lambda$ means that the iteration stops when the gradient is zero or rather small.

4 Simulations

Simulations were performed on a partially coherent imaging system with a quasar source illumination (σ_{out}/σ_{in}/degree = 0.9/0.6/45 deg). The wavelength λ in the simulations was set at 193 nm and NA was 1.35. The resist effect was approximated by a Sigmoid function with $a = 100$ and $t = 0.4$. The mask quadratic error regularization factor β_1 and the mask complexity regularization factor β_2 were both set at 0.01. The parameter κ_1 and κ_2 of the initial mask in Eq. (29) were 0.90 and 0.05, respectively. The value of Λ was set at 0.3 in termination criterion $\|v^{k+1}\| < \Lambda$. Instead of computing the step length γ^k in Eq. (34) accurately, we directly set γ^k at a small constant 0.3. All the simulations were carried out with in-house MATLAB® codes on an HPZ820 Workstation (2×2 GHz Intel Xeons with 8Cores/16Threads) using a Windows 7 (64 bit) operating system.

4.1 Results of Robust Inverse Lithography

Figure 3 depicts the synthesized mask patterns by nominal and robust ILT. Figure 3(a) shows the desired pattern consisting of 361×361 pixels with a grid size of 2.5 nm, which is commonly encountered in logic circuit design. From Fig. 3, the synthesized mask patterns by robust ILT are quite different from that by nominal ILT. It is particularly interesting to note that subresolution assist features (SRAFs) are generated in Fig. 3(c), but not in Fig. 3(b). The reason might be that defocus from the nominal condition results in much worsened image contrast; thus SRAFs tend to be produced in order to enhance the image quality.

Moreover, we compare the exposure-defocus (E-D) trees of the synthesized mask patterns by setting the EDE to within ±10% of the CD target (45 nm in this case) in Fig. 4. As expected, the synthesized mask patterns by robust ILT have wider E-D windows. This is because the robust ILT further takes the process variations into consideration. As shown in Figs. 3 and 4, the robust ILT tends to generate SRAFs and has the capability of achieving wider E-D windows compared to that synthesized for the nominal condition only.

4.2 Results of Cascadic Multigrid Algorithm

In this case, we set three level spaces in total, Ω_1, Ω_2, and Ω_3, with the corresponding grid size of 15, 5, and 2.5 nm. A bilinear method is used to interpolate from a coarse grid space to a finer one. Figure 5 depicts the simulated images of the CMG algorithm. Since the grid size on the space Ω_1 is very coarse, the synthesized mask pattern, Fig. 5(b), on the
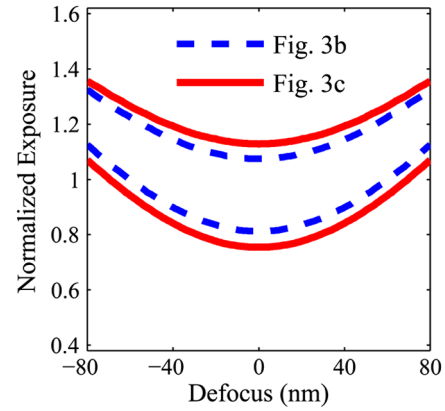

Fig. 4 Exposure-defocus (E-D) trees of the synthesized mask patterns [(b) and (c)] obtained, respectively, by nominal and robust ILT.

space Ω_1 reveals a general pattern and consists of numerous gray-level SRAFs. These general pattern and gray-level SRAFs are difficult to manufacture in practice and require further correction on the fine spaces. Compared to the mask pattern synthesized by the conventional method [Fig. 3(c)], the mask pattern synthesized by CMG algorithm [Fig. 5(f)] is mostly the same but possesses more SRAFs.

Figure 6(a) illustrates the convergence comparison for the statistical EDE. The CMG algorithm takes 132 iterations to find an optimal mask pattern, Fig. 5(b), with a statistical EDE of 4.20 nm, and this mask pattern is then interpolated to serve as an initial value on the space Ω_2, as shown in Fig. 5(c). It is noted that the convergence history jumps to 5.34 nm here [shown in the blue circle in Fig. 6(a)]; this is because the synthesized coarse mask pattern [Fig. 5(b)] is not exactly accurate and, thus, required further correction. The CMG algorithm takes further 87 iterations on space Ω_2 and reaches a mask pattern, Fig. 5(d), with a statistical EDE of 4.44 nm. Finally, the CMG algorithm achieves a mask pattern, Fig. 5(f), after 20 iterations on the finest space Ω_3. From Fig. 6(a), it is observed that the CMG algorithm achieves a smaller statistical EDE of 4.46 nm compared to 4.51 nm by the conventional method, which directly synthesizes mask on the finest space Ω_3. Moreover, we also depict the convergence of pattern error in the CMG algorithm. It is observed that pattern error changed significantly when mask is interpolated from one grid space to another, and it is, thus, not intuitive to study on the convergence. Since pattern error is a dimensionless quantity and highly depends on mask feature and simulation parameters (such as simulation grid resolution), it is not suited for the CMG algorithm.
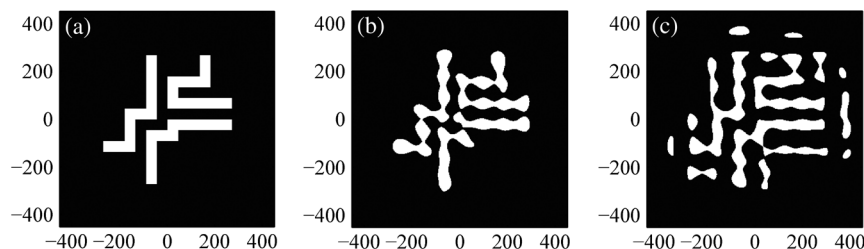
Fig. 3 (a) Desired pattern. (b) The synthesized mask pattern by nominal inverse lithography technology (ILT). (c) The synthesized mask pattern by robust ILT. The horizontal axis and vertical axis denote x and y position of the patterns in nanometers, respectively.
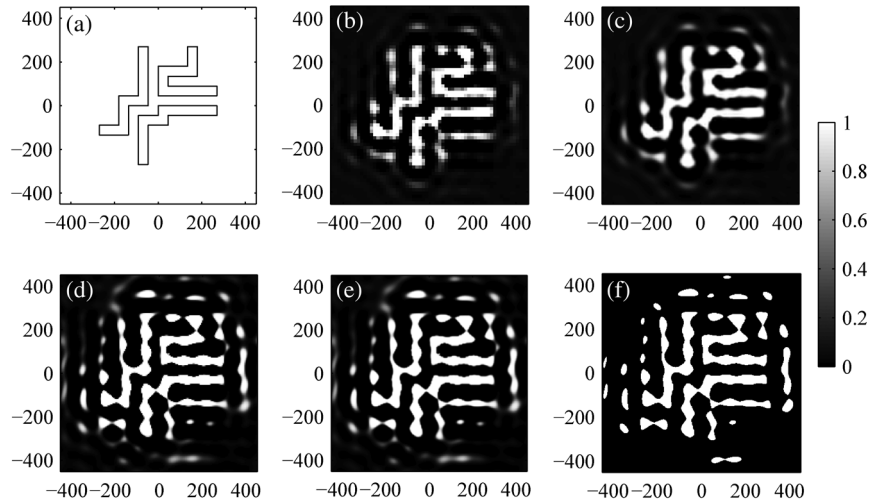
Fig. 5 (a) Desired pattern. (b) The synthesized mask pattern on the coarsest space Ω_1. (c) The interpolated mask pattern of (b) from the space Ω_1 to the space Ω_2. (d) The synthesized mask pattern on the space Ω_2 with an initial value (c). (e) The interpolated mask of (d) from the space Ω_2 to the space Ω_3. (f) The synthesized mask pattern on the finest space Ω_3. The horizontal axis and vertical axis denote x and y position of the patterns in nanometers, respectively.

The runtime of the conventional method and the CMG algorithm on each space is shown in Table 2. Since the optimization variables (grids) are relatively less on the space Ω_1, it synthesizes an optimal mask pattern, Fig. 5(b), after 8.67 s compared to 620.03 s by directly synthesizing mask on the finest space Ω_3. Overall, the CMG algorithm takes 239 iterations using 135.35 s compared to 193 iterations using 620.03 s by the conventional method; in other words, the CMG algorithm is 4.6× faster than the conventional method although taking more iterations. Moreover, we compare the E-D windows for the synthesized mask pattern by the conventional method and the CMG algorithm in Fig. 6(b). It is observed that the synthesized mask pattern by the CMG algorithm has slightly improved the E-D window. The reason might be that CMG algorithm generates more SRAFs.

Since there are rather less optimization variables 61×61 grids on the space Ω_1 than 361×361 grids on the space Ω_3, it takes much less runtime and is more likely to change each variable to find a better solution. As expected, the synthesized result on the coarsest space Ω_1 contains a general pattern with numerous SRAFs. From the perspective of signal processing, these patterns can be considered low-frequency portions in mask, and these patterns are further corrected by adding high-frequency details on the fine space. The conventional method accomplishes mask synthesis directly on the fine space Ω_3; it, therefore, is expensive to change each variable and cannot handle the low-frequency portions in mask explicitly. As a result, the CMG algorithm has the capacity of achieving a mask pattern with smaller statistical EDE and wider E-D window compared to the conventional method.
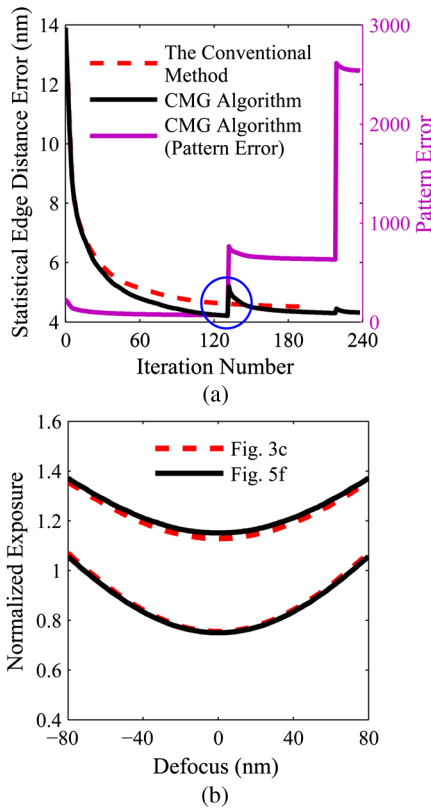

Fig. 6 (a) Convergence history of the conventional method and cascadic multigrid (CMG) algorithm. (b) E-D trees of the synthesized mask patterns [(c) and (f)] obtained, respectively, by the conventional method and CMG algorithm.

Table 2 Runtime of the conventional method and CMG algorithm on each space for a desired pattern Fig. 5(a).
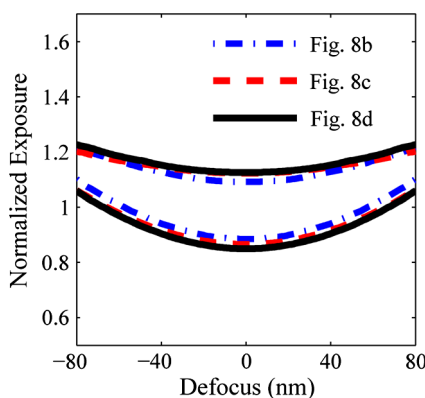
	CMG algorithm				Conventional method
	15 nm	5 nm	2.5 nm	Total	
Iteration num.	132	87	20	239	193
Time (s)	8.67	62.41	64.27	135.35	620.03
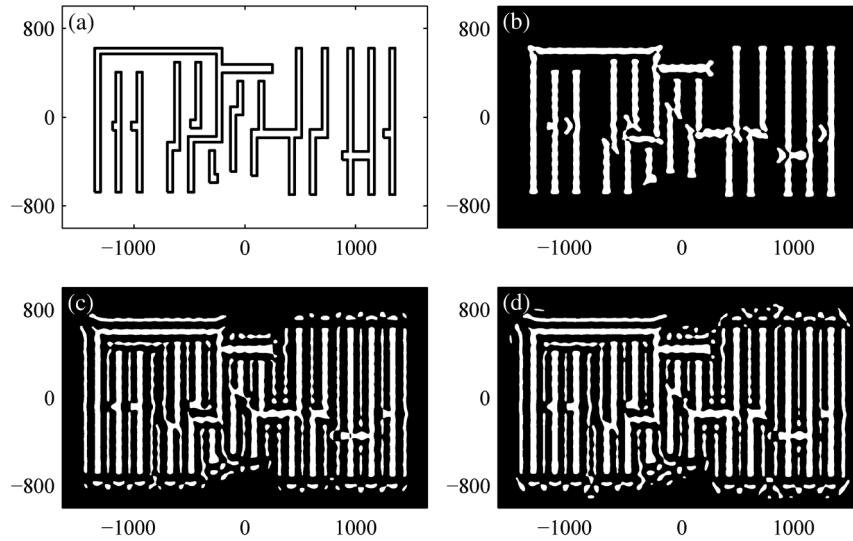
Fig. 7 (a) Desired pattern. The synthesized mask pattern obtained by (b) nominal ILT, (c) robust ILT using the conventional method, and (d) robust ILT using the CMG algorithm. The horizontal axis and vertical axis denote x and y position of the patterns in nanometers, respectively.

We also performed simulations for a more complicated pattern in Fig. 7. Figure 7(a) shows the desired pattern, which is a metal layer of the benchmark clock gate circuit pattern within a simulation area of $2\ \mu m \times 3.3\ \mu m$.[44] As expected, the synthesized mask pattern, Fig. 7(b), by nominal ILT is quite different from that synthesized by robust ILT, Figs. 7(c) and 7(d), and the robust ILT has significantly improved the E-D window (see Fig. 8). On the other hand, compared to the synthesized mask pattern by the conventional method, CMG algorithm reaches a mostly same pattern but with more SRAFs; the E-D window has, thus, slightly improved. The runtime of the conventional method and the CMG algorithm on each space is shown in Table 3. It is revealed that the CMG algorithm is overall 4.3× faster than the conventional method. This set of simulations further demonstrates that the CMG algorithm has big improvement in computation efficiency, along with a slightly better lithographic performance (wider E-D window).

Table 3 Runtime of the conventional method and CMG algorithm on each space for a desired pattern Fig. 7(a).

	CMG algorithm				Conventional method
	15 nm	5 nm	2.5 nm	Total	
Iteration num.	154	106	22	282	247
Time (s)	97.03	735.64	628.79	1.48×10^3	6.37×10^3

5 Conclusions

In this paper, we report a CMG method for robust inverse mask synthesis. This method synthesizes mask hierarchically. It first computes an initial mask pattern on a coarse grid and, therefore, spends less runtime and is more likely to change each variable to find a better solution. This initial mask is then interpolated to a fine grid and to be further corrected, which requires a small number of iterations on the fine grid. Overall, the CMG algorithm is able to run more than four times faster than conventional mask synthesis methods working on a fixed fine grid. In addition, we apply an analytical circle-sampling technique to the forward lithography imaging modeling and employ a novel EDE metric to guide mask synthesis. These two techniques are both independent of grid size and are well suited for the CMG algorithm. It is expected that the proposed method will provide a useful mask synthesis strategy in practical optical lithography.

Fig. 8 E-D trees of the synthesized mask patterns [(b) to (d)] obtained, respectively, by nominal ILT using the conventional method, robust ILT using the conventional method, and robust ILT using the CMG algorithm.

Appendix: Derivation of Eq. (25)

The gradient of the cost function $G(M)$ with respect to mask M is expressed in Eq. (25). In the following, we present the detailed expression of each term in Eq. (25), i.e., $\nabla_M \text{EDE}_s(M)$, $\nabla_M R_Q(M)$, and $\nabla_M R_{\text{TV}}(M)$. In the present content, $\mathbf{r}$ and ρ denote the spatial coordinate (x, y).

According to Ref. 28, with a fixed defocus h_m and dose variation t_n, the gradient of EDE with respect to mask M is

$$\nabla_M \text{EDE}(M; h_m, t_n) = \frac{\partial \text{EDE}(M; h_m, t_n)}{\partial M} = a \cdot \frac{\delta_x \cdot \delta_y}{L}$$

$$\times \left\{ \sum_{i=1}^{Q} \mu_i \phi_i^{\text{flip}}(\mathbf{r}; h_m) \otimes [(Z - Z^*) \cdot Z \right.$$

$$\cdot (1 - Z) \cdot ([\phi_i(\mathbf{r}; h_m)]^\dagger \otimes M)] \right\}$$

$$+ a \cdot \frac{\delta_x \cdot \delta_y}{L} \left\{ \sum_{i=1}^{Q} [\mu_i \phi_i^{\text{flip}}(\mathbf{r}; h_m)]^\dagger \right.$$

$$\otimes [(Z - Z^*) \cdot Z \cdot (1 - Z)$$

$$\left. \cdot (\phi_i(\mathbf{r}; h_m) \otimes M)] \right\}, \tag{40}$$

where $\phi_i(\mathbf{r}; h)$ is the i'th optical kernels with the defocus h and Z is the corresponding resist image of input mask M under the defocus h_m and dose variation t_n. So the gradient of the statistical EDE with respect to mask M is

$$\nabla_M \text{EDE}_s(M) = \frac{\partial \sum_m^{N_h} \sum_n^{N_t} [\alpha(h_m) \cdot \zeta(t_n) \cdot \text{EDE}(M; h_m, t_n)]}{\partial M}$$

$$= \sum_m^{N_h} \sum_n^{N_t} \left[\alpha(h_m) \cdot \zeta(t_n) \cdot \frac{\partial \text{EDE}(M; h_m, t_n)}{\partial M} \right]$$

$$= \sum_m^{N_h} \sum_n^{N_t} [\alpha(h_m) \cdot \zeta(t_n) \cdot \nabla_M \text{EDE}(M; h_m, t_n)]. \tag{41}$$

The gradient of the mask quadratic error $R_Q(M)$ with respect to mask M is

$$\nabla_M R_Q(M) = \frac{\partial \int_\Omega \{1 - [2M(\mathbf{r}) - 1]^2\} d\mathbf{r}}{\partial M(\rho)}$$

$$= \frac{\partial \sum_{\mathbf{r}} \{-4M^2(\mathbf{r}) + 4M(\mathbf{r})\}}{\partial M(\rho)} = -8M + 4. \tag{42}$$

The gradient of the mask complexity $R_{\text{TV}}(M)$ with respect to mask M is

$$\nabla_M R_{\text{TV}}(M) = \frac{\partial \int_\Omega (|\nabla M_x| + |\nabla M_y|) d\mathbf{r}}{\partial M(\rho)}$$

$$= \frac{\partial \sum_{\mathbf{r}} [|YM(\mathbf{r})| + |M(\mathbf{r}) Y^T|]}{\partial M(\rho)}$$

$$= Y^T \text{sign}[YM] + \text{sign}[MY^T] Y, \tag{43}$$

where $\text{sign}[\cdot]$ is the Signum function, and Y is an operator of the first derivative as

$$Y = \begin{bmatrix} 1 & -1 & & & 0 \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ 0 & & & -1 & 1 \end{bmatrix}. \tag{44}$$

References

1. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, SPIE Press, Bellingham, Washington (2001).
2. L. W. Liebmann et al., "TCAD development for lithography resolution enhancement," *IBM J. Res. Dev.* **45**(5), 651–665 (2001).
3. F. Schellenberg, "Resolution enhancement technology: the past, the present, and extensions for the future," *Proc. SPIE* **5377**, 1–20 (2004).
4. L. Pang, Y. Liu, and D. Abrams, "Inverse lithography technology (ILT): what is the impact to the photomask industry?," *Proc. SPIE* **6283**, 62830X (2006).
5. L. Pang, Y. Liu, and D. Abrams, "Inverse lithography technology (ILT): a natural solution for model-based SRAF at 45 nm and 32 nm," *Proc. SPIE* **6607**, 660739 (2007).
6. D. O. S. Melville et al., "Computational lithography: exhausting the resolution limits of 193-nm projection lithography systems," *J. Vac. Sci. Technol. B* **29**(6), 06FH04 (2011).
7. M. L. Rieger, "Communication theory in optical lithography," *J. Micro/ Nanolith. MEMS MOEMS* **11**(1), 013003 (2012).
8. A. Villaret et al., "Inverse lithography technique for advanced CMOS nodes," *Proc. SPIE* **8683**, 86830E (2013).
9. T. Cecil et al., "Enhancing fullchip ILT mask synthesis capability for IC manufacturability," *Proc. SPIE* **7973**, 79731C (2011).
10. S. S. Yeh et al., "Study of the pattern aware OPC," *Proc. SPIE* **9052**, 90521Y (2014).
11. I. Torunoglu et al., "A GPU-based full-chip inverse lithography solution for random patterns," *Proc. SPIE* **7641**, 764115 (2010).
12. Y. Liu and A. Zakhor, "Optimal binary image design for optical lithography," *Proc. SPIE* **1264**, 401–412 (1990).
13. Y. Liu and A. Zakhor, "Binary and phase shifting mask design for optical lithography," *IEEE Trans. Semiconduct. Manuf.* **5**(2), 138–152 (1992).
14. Y. C. Pati and T. Kailath, "Phase-shifting masks for microlithography: automated design and mask requirements," *J. Opt. Soc. Am. A* **11**(9), 2438–2452 (1994).
15. S. Sherif, B. Saleh, and R. D. Leone, "Binary image synthesis using mixed linear integer programming," *IEEE Trans. Image Process.* **4**(9), 1252–1257 (1995).
16. Y. Granik, "Solving inverse problems of optical microlithography," *Proc. SPIE* **5754**, 506–526 (2005).
17. Y. Granik, "Fast pixel-based mask optimization for inverse lithography," *J. Micro/Nanolith. MEMS MOEMS* **5**(4), 043002 (2006).
18. A. Poonawala and P. Milanfar, "Mask design for optical microlithography—an inverse imaging problem," *IEEE Trans. Image Process.* **16**(3), 774–788 (2007).
19. A. Poonawala and P. Milanfar, "A pixel-based regularization approach to inverse lithography," *Microelectron. Eng.* **84**(12), 2837–2852 (2007).
20. X. Ma and G. R. Arce, "Generalized inverse lithography methods for phase shifting mask design," *Opt. Express* **15**(23), 15066–15079 (2007).
21. S. H. Chan, A. K. Wong, and E. Y. Lam, "Initialization for robust inverse synthesis of phase-shifting masks in optical projection lithography," *Opt. Express* **16**(19), 14746–14760 (2008).
22. X. Ma and G. R. Arce, "Binary mask optimization for inverse lithography with partially coherent illumination," *J. Opt. Soc. Am. A* **25**(12), 2960–2970 (2008).
23. E. Y. Lam and A. K. Wong, "Computation lithography: virtual reality and virtual virtuality," *Opt. Express* **17**(15), 12259–12268 (2009).
24. J. C. Yu and P. Yu, "Impacts of cost functions on inverse lithography patterning," *Opt. Express* **18**(22), 23331–23342 (2010).
25. X. Ma and G. R. Arce, "Pixel-based OPC optimization based on conjugate gradients," *Opt. Express* **19**(3), 2165–2180 (2011).

26. J. Li, Y. Shen, and E. Y. Lam, "Hotspot-aware fast source and mask optimization," *Opt. Express* **20**(19), 21792–21804 (2012).
27. J. Li, S. Y. Liu, and E. Y. Lam, "Efficient source and mask optimization with augmented Lagrangian methods in optical lithography," *Opt. Express* **21**(7), 8076–8090 (2013).
28. W. Lv et al., "Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step," *J. Vac. Sci. Technol. B* **31**(4), 041605 (2013).
29. W. Lv, Q. Xia, and S. Y. Liu, "Pixel-based inverse lithography using a mask filtering technique," *Proc. SPIE* **8683**, 868325 (2013).
30. W. Lv, Q. Xia, and S. Y. Liu, "Mask-filtering-based inverse lithography," *J. Micro/Nanolith. MEMS MOEMS* **12**(4), 043003 (2013).
31. N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *J. Opt.* **12**(4), 045601 (2010).
32. Y. Shen et al., "Robust level-set-based inverse lithography," *Opt. Express* **19**(6), 5511–5521 (2011).
33. N. Jia and E. Y. Lam, "Pixelated source mask optimization for process robustness in optical lithography," *Opt. Express* **19**(20), 19384–19398 (2011).
34. S. K. Choy et al., "A robust computational algorithm for inverse photomask synthesis in optical projection lithography," *SIAM J. Imaging Sci.* **5**(2), 625–651 (2012).
35. S. Li, X. Wang, and Y. Bu, "Robust pixel-based source and mask optimization for inverse lithography," *Opt. Laser Technol.* **45**, 285–293 (2013).
36. S. Y. Liu et al., "Convolution-variation separation method for efficient modeling of optical lithography," *Opt. Lett.* **38**(13), 2168–2170 (2013).
37. F. A. Bornemann and P. Deuflhard, "The cascadic multigrid method for elliptic problems," *Numer. Math.* **75**(2), 135–152 (1996).
38. D. Braess, P. Deuflhard, and K. Lipnikov, "A subspace cascadic multigrid method for mortar elements," *Computing* **69**(3), 205–225 (2002).
39. W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM Press, Philadelphia, PA (2000).
40. L. Reichel and A. Shyshkov, "Cascadic multilevel methods for ill-posed problems," *J. Comput. Appl. Math.* **233**(5), 1314–1325 (2010).
41. P. Gong et al., "Fast aerial image simulations for partially coherent systems by transmission cross coefficient decomposition with analytical kernels," *J. Vac. Sci. Technol. B* **30**(6), 06FG03 (2012).
42. B. E. A. Saleh, "Optical bilinear transformations: general properties," *Opt. Acta* **26**(6), 777–799 (1979).
43. A. K. Wong, *Optical Imaging in Projection Microlithography*, SPIE Press, Bellingham, Washington (2005).
44. "The NanGate 45nm Open Cell Library, v1.3," http://www.si2.org/openeda.si2.org/projects/nangatelib (1 July 2013).

Wen Lv is currently a PhD candidate at Huazhong University of Science and Technology under the guidance of Professor Shiyuan Liu. He received his BS degree from the School of Mechanical Science and Engineering of the same university in 2011. His research involves various issues in optical lithography, including inverse lithography, fast optical image simulation, and mask writing technique. He is a student member of SPIE and IEEE.

Edmund Y. Lam received his BS (with distinction), MS, and PhD degrees in electrical engineering from Stanford University. He was a senior engineer at KLA-Tencor Corporation in San Jose, California, before joining the Electrical and Electronic Engineering Department at the University of Hong Kong, where he is currently an associate professor. He is also the founding director of its Imaging Systems Laboratory. His research interests include computational optics and imaging, particularly their applications in the semiconductor manufacturing process. He is a fellow of SPIE and OSA.

Haiqing Wei received his BS in physics from Peking University, China, MS in physics, and PhD in electrical and computer engineering from McGill University, Canada. He has been a research scientist and engineer for over a decade, working in various subjects of fiber-optic communications, optical imaging, nanophotonics, and computational lithography. He has over 20 patent applications in these subject areas and has published over 30 scientific articles in peer-reviewed journals. Since April 2012, he has been appointed as a guest professor at Huazhong University of Science and Technology.

Shiyuan Liu is a professor of mechanical engineering at Huazhong University of Science and Technology, leading his Nanoscale and Optical Metrology Group with research interest in metrology and instrumentation for nanomanufacturing. He also actively works in the area of optical lithography, including partially coherent imaging theory, wavefront aberration metrology, optical proximity correction, source mask optimization, and inverse lithography technology. He received his PhD in mechanical engineering from Huazhong University of Science and Technology in 1998. He is a member of SPIE, OSA, AVS, IEEE, and Chinese Society of Micro/Nano Technology. He holds 21 patents and has authored or coauthored more than 100 technical papers.