<table>
<thead>
<tr>
<th>Title</th>
<th>Cascadic multigrid algorithm for robust inverse mask synthesis in optical lithography</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Lv, W; Lam, EYM; Wei, H; Liu, S</td>
</tr>
<tr>
<td>Citation</td>
<td>Journal of Micro/Nanolithography, MEMS, and MOEMS, 2014, v. 13 n. 2, p. 023003-1-023003-10</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2014</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/200618">http://hdl.handle.net/10722/200618</a></td>
</tr>
<tr>
<td>Rights</td>
<td>Journal of Micro/Nanolithography, MEMS, and MOEMS. Copyright © SPIE - International Society for Optical Engineering.; Copyright notice format: Copyright 2014 Society of Photo-Optical Instrumentation Engineers. One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.; This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
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, a 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, State Key Laboratory of Digital Manufacturing Equipment and Technology, 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]

Keywords: robust inverse mask synthesis; optical lithography; cascadic multigrid algorithm; analytical circle-sampling technique; edge distance error.

Paper 14007 received Jan. 22, 2014; revised manuscript received Mar. 26, 2014; accepted for publication Mar. 28, 2014; published online Apr. 28, 2014.

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.4-7 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 masks20,21 and partially coherent imaging systems,22-26 and the optimization algorithm was improved with an active set method21 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 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
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 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 partial coherent systems by decomposing the double-impulse response (DIR) function into analytical kernels. The DIR function, whose Fourier transform is the familiar transmission cross-coefficient (TCC), is projected onto the circle-sampling function (CSF) space and converted into a circular function, whose Fourier transform is the familiar transmission cross-coefficient (TCC), is projected onto the circle-sampling function (CSF) space and converted into a circular function, whose Fourier transform is the familiar transmission cross-coefficient (TCC). Here, \(H(r_1)\) is the point spread function (PSF) of the optical system that may be obtained by an inverse Fourier transform of the pupil function \(H(f)\) as

\[
H(r) = \text{IFT}\{H(f)\},
\]

where \(\text{IFT}\{\cdot\}\) is the inverse Fourier transform. Since the DIR function is Hermitian and band-limited, it is possible to approximate the DIR function as a finite CSF series as

\[
D(r_1, r_2) = \sum_{k=1}^{N} \sum_{l=1}^{M} p_{k,l} \varphi_k(r_1) \varphi_l^*(r_2),
\]

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

\[
P = \sum_{i=1}^{Q} \mu_i s_i s_i^H.
\]

where \(s_i\) is the \(i\)th eigenvector with \(Q\) eigenvectors in total, \(s_i^H\) is its Hermitian conjugate, and \(\mu_i\) is the corresponding eigenvalue. Substituting Eq. (5) into Eq. (4), we can analytically express the DIR function as

\[
D(r_1, r_2) = \sum_{i=1}^{Q} \mu_i \phi_i(r_1) \phi_i^*(r_2),
\]

where \(\phi_i(r)\) is called the \(i\)th analytical optical kernel, which is calculated as

\[
\phi_i(r) = \sum_{l=1}^{N} s_{i,l} \varphi_l(r).
\]

where \(s_{i,l}\) is the \(l\)th element of \(s_i\). 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

\[
I(r) = \int \int D(r - r_1, r - r_2) M(r_1) M^*(r_2) dr_1 dr_2,
\]
flow chart of the double-impulse response function decomposition into analytical kernels.

\[ I(r) = \sum_{i=1}^{Q} \mu_i |\phi_i(r) \otimes M(r)|^2. \]  

Subsequently, the optical image \( I(r) \) goes through the resist development to form the printed image on the wafer. The resist effect is often approximated by a constant threshold level of the resist.

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

\[ Z(r) = \Gamma\{M(r)\} = \text{sig}[I(r)] \]
\[ = \text{sig}\left[ \sum_{i=1}^{Q} \mu_i |\phi_i(r) \otimes M(r)|^2 \right]. \]

where the operator \( \Gamma\{\cdot\} \) implements the forward mapping from the input mask \( M(r) \) to the output pattern \( Z(r) \) 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. 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, \( S_{\text{shadow}} \) 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 \( \delta_x \) and \( \delta_y \) are the lengths of the element along the \( x \) and \( y \) directions, respectively. EDE is defined as

\[ \text{EDE} = \frac{S_{\text{shadow}}}{L}. \]

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

\[ S_{\text{shadow}} = N_s \cdot (\delta_x \cdot \delta_y). \]

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

\[ N_s = \|\Gamma\{M\} - Z^*\|^2_2. \]

where \( \| \cdot \|^2_2 \) 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

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

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

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

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

where $\nabla M_x = \partial M/\partial x$, $\nabla M_y = \partial M/\partial y$, and $\Omega$ 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 $\beta_1$ and $\beta_2$ to be stated as one single optimization objective as

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

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

$$M^* = \arg \min_M F(M).$$

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 $H(r, h)$,

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

where $j = \sqrt{-1}$, $NA$ is the numerical aperture, and $\lambda$ 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

$$EDE_s(M) = \frac{N_h}{N_m} \sum_{m} \sum_{n} [\alpha(h_m) \cdot \zeta(t_n) \cdot EDE(M; h_m, t_n)],$$

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. 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

$$G(M) = EDE_s(M) + \beta_1 R_Q(M) + \beta_2 R_{TV}(M).$$

The robust inverse lithography problem is, thus, defined as

$$M^* = \arg \min_M G(M).$$

### 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

$$\Omega_1, \Omega_2, \ldots, \Omega_C,$$

with the corresponding space grid size

$$\delta_1 > \delta_2 > \ldots > \delta_C.$$

Since the grid is usually square, and we use only a parameter $\delta$ to describe the space grid size for simplicity, that means $\delta = \delta_1 = \delta_2 = \ldots = \delta_C$. The CMG algorithm starts mask synthesis on the coarsest space $\Omega_1$ and obtains a mask $M_1$. The obtained mask $M_1$ is then embedded in the next space $\Omega_2$, and the interpolation operation, and then the interpolated mask $M_2$ serves as the initial value on the space $\Omega_2$. With the initial value $M_2$, it calculates a mask $M_3$ on the space $\Omega_3$. The computation repeats in this manner untill a mask $M_C$ on the finest space $\Omega_C$ is obtained. The pseudo-code of CMG algorithm is described in Table 1.

The CMG algorithm starts mask synthesis on the coarsest space $\Omega_1$. Since the coarsest mask contains much less optimization variables (grids) than the finest mask on space $\Omega_C$, 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 $\Omega_C$, 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.
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

$$
\nabla G(M) = \nabla \text{Vol}(M) + \beta_1 \nabla R_{0}(M) + \beta_2 \nabla R_{TV}(M).
$$

(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

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

(26)

Here, $g^k$ is a matrix representing the gradient $\nabla G(M)$ of cost function $G(M)$ at mask $M$, and $\eta^k$ 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-Ribiére-Polyak (PRP) CG method. 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 $\eta^k$ to avoid jamming. It essentially performs a restart when a bad direction occurs. In PRP CG method, $\eta^k$ is defined as

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

(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:

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

(28)

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

$$
M^0(\mathbf{r}) = k_1 \cdot Z^*(\mathbf{r}) + k_2.
$$

(29)

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

(30)

where $k_1$ and $k_2$ are parameters to adjust the initial mask value; for example, $k_1 = 0.90$ and $k_2 = 0.05$ 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. Finally, we calculate the initial gradient $g^0$ and the optimization direction $v^0$ based on Eq. (26).

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

(31)

$$
v^0 = -g^0,
$$

(32)

where

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

(33)

**Iteration $k$:**

**Step i:** Search the step length $r^k$ in the direction $v^k$,

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

(34)

**Step ii:** Update $\theta^{k+1}$,

$$
\theta^{k+1} = \theta^k + r^k \cdot v^k.
$$

(35)

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

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

(36)

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

(37)

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

(38)

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

**Stop:** Obtain the mask.

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

(39)

In the above procedure, the iteration is terminated when $\|v^{k+1}\| < \Lambda$ or $k > \Psi$, where $\Lambda$ is defined as the minimum value of the norm of velocity and $\Psi$ 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 ($\sigma_{\text{out}}/\sigma_{\text{in}}/\text{degree} = 0.9/0.6/45 \text{ deg}$). The wavelength $\lambda$ 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 $\beta_1$ and the mask complexity regularization factor $\beta_2$ were both set at 0.01. The parameter $k_1$ and $k_2$ of the initial mask in Eq. (29) were 0.90 and 0.05, respectively. The value of $\Lambda$ was set at 0.3 in termination criterion $\|x^{k+1}\| < \Lambda$. Instead of computing the step length $\gamma^k$ in Eq. (34) accurately, we directly set $\gamma^k$ at a small constant 0.3. All the simulations were carried out with in-house MATLAB® codes on an HPZ820 Workstation (2 x 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 x 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, $\Omega_1$, $\Omega_2$, and $\Omega_3$, 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 $\Omega_1$ is very coarse, the synthesized mask pattern, Fig. 5(b), on the space $\Omega_1$ 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 $\Omega_2$, 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 $\Omega_2$ 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 $\Omega_3$. 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 $\Omega_3$. 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](Image 365x599 to 525x750)

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. 4](Image 154x95 to 474x185)

Fig. 4 Exposure-defocus (E-D) trees of the synthesized mask patterns [(b) and (c)] obtained, respectively, by nominal and robust ILT.
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 $\Omega_1$, 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 $\Omega_3$. 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 $4.6 \times$ 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 $61 \times 61$ grids on the space $\Omega_1$ than $361 \times 361$ grids on the space $\Omega_3$, 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 $\Omega_1$ 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 $\Omega_3$; 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.

**Table 2** Runtime of the conventional method and CMG algorithm on each space for a desired pattern Fig. 5(a).  

<table>
<thead>
<tr>
<th></th>
<th>15 nm</th>
<th>5 nm</th>
<th>2.5 nm</th>
<th>Total</th>
<th>Conventional method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration num.</td>
<td>132</td>
<td>87</td>
<td>20</td>
<td>239</td>
<td>193</td>
</tr>
<tr>
<td>Time (s)</td>
<td>8.67</td>
<td>62.41</td>
<td>64.27</td>
<td>135.35</td>
<td>620.03</td>
</tr>
</tbody>
</table>
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 \(2 \mu m \times 3.3 \mu m\). 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 4.3x 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

<table>
<thead>
<tr>
<th></th>
<th>CMG algorithm</th>
<th>Conventional method</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 nm</td>
<td>154</td>
<td>247</td>
</tr>
<tr>
<td>5 nm</td>
<td>106</td>
<td>628.79</td>
</tr>
<tr>
<td>2.5 nm</td>
<td>22</td>
<td>1.48 \times 10^3</td>
</tr>
<tr>
<td>Total</td>
<td>282</td>
<td>6.37 \times 10^3</td>
</tr>
</tbody>
</table>

### 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.

### Acknowledgments

This work was funded by the National Natural Science Foundation of China (Grant No. 91023032, 51005091, 51121002), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120142110019), the National Science and
Technology Major Project of China (Grant No. 2012ZX02701001), and the National Instrument Development Specific Project of China (Grant No. 2011YQ160002). It was also supported in part by the UGC Areas of Excellence project Theory, Modeling, and Simulation of Emerging Electronics, and by the State Key Lab of Digital Manufacturing Equipment and Technology under Project DMETKF2013003.

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., $V_M EDE(M)$, $V_M R_Q(M)$, and $V_M R_{TV}(M)$. In the present content, $r$ and $\rho$ 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

$$V_M EDE(M; h_m, t_n) = \frac{\partial EDE(M; h_m, t_n)}{\partial M} = a \cdot \delta_x \cdot \delta_y \cdot L \cdot \left\{ \sum_{i=1}^{\Omega} \mu_i \phi_i^0 \cdot (r; h_m) \otimes [(Z - Z^*) \cdot Z \cdot (1 - Z) \cdot \{(\phi_i; r; h_m) \otimes M)] \ight. \\
\left. + a \cdot \delta_x \cdot \delta_y \cdot \left\{ \sum_{i=1}^{\Omega} \mu_i \phi_i^0 \cdot (r; h_m) \otimes [(Z - Z^*) \cdot Z \cdot (1 - Z) \cdot \{(\phi_i; r; h_m) \otimes M)] \right\} \right. \\
\cdot \left\{ \sum_{i=1}^{\Omega} \mu_i \phi_i^0 \cdot (r; h_m) \otimes M \right\},$$

where $\phi_i(r; h)$ is the $i$’th optical kernel 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

$$V_M EDE_s(M) = \sum_{n=1}^{N_s} \sum_{m=1}^{N_m} \left[ \alpha(h_m) \cdot \zeta(t_n) \cdot \frac{\partial EDE(M; h_m, t_n)}{\partial M} \right]$$

$$= \sum_{m=1}^{N_m} \sum_{n=1}^{N_n} \left[ \alpha(h_m) \cdot \zeta(t_n) \cdot \frac{\partial EDE(M; h_m, t_n)}{\partial M} \right]$$

$$= \sum_{m=1}^{N_m} \sum_{n=1}^{N_n} \left[ \alpha(h_m) \cdot \zeta(t_n) \cdot V_M EDE(M; h_m, t_n) \right].$$

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

$$V_M R_Q(M) = \frac{\partial}{\partial M} \int_O \left[1 - |2M(r) - 1|^2 \right] dr$$

$$= \sum_{r} \left[ -4M^2(r) + 4M(r) \right] = -8M + 4.$$ 

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

$$V_M R_{TV}(M) = \frac{\partial}{\partial M} \int_O \left[ |\nabla M| + |\nabla M| \right] dr$$

$$= \sum_{r} \left[ -4 |\nabla M(r) | + 4 |\nabla M(r) | \right] = -8M + 4.$$ 

\[ \text{(43)} \]

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

$$Y = \begin{bmatrix}
1 & -1 & 0 \\
1 & -1 & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1 & -1 & 0
\end{bmatrix}. \text{(44)}$$

References