Title	Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step
Author(s)	Lv, W; Liu, S; Xia, Q; Wu, X; Shen, Y; Lam, EYM
Citation	Journal of Vacuum Science and Technology: Part B Nanotechnology & Microelectronics, 2013, v. 31 n. 4, p. 041605-1-041605-13
Issued Date	2013
URL	http://hdl.handle.net/10722/200624
Rights	Copyright 2013 American Vacuum Society. This article may be downloaded for personal use only. Any other use requires prior permission of the author and the American Vacuum Society. The following article appeared in Journal of Vacuum Science and Technology: Part B Nanotechnology & Microelectronics, 2013, v. 31 n. 4, p. 041605-1-041605-13 and may be found at http://scitation.aip.org/content/avs/journal/jvstb/31/4/10.1116/1.4813781

Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step

Wen Lv, Shiyuan Liu, Qi Xia, Xiaofei Wu, Yijiang Shen, and Edmund Y. Lam

Articles you may be interested in

Stitching periodic submicron fringes by utilizing step-and-align interference lithography
J. Vac. Sci. Technol. B **27**, 2951 (2009); 10.1116/1.3258152

GaN-based light-emitting diodes with indium tin oxide texturing window layers using natural lithography
Appl. Phys. Lett. **86**, 221101 (2005); 10.1063/1.1940723

Prediction of fabrication distortions in step and flash imprint lithography templates
J. Vac. Sci. Technol. B **20**, 2891 (2002); 10.1116/1.1521743

Nanofabrication using a stencil mask
Appl. Phys. Lett. **75**, 1631 (1999); 10.1063/1.124777

Lithography with a mask of block copolymer microstructures
J. Vac. Sci. Technol. B **16**, 544 (1998); 10.1116/1.589860

Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step

Wen Lv
Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China

Shiyuan Liu[a] and Qi Xia
State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Xiaofei Wu, Yijiang Shen, and Edmund Y. Lam
Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

Inverse mask synthesis is achieved by minimizing a cost function on the difference between the output and desired patterns. Such a minimization problem can be solved by a level-set method where the boundary of the pattern is iteratively evolved. However, this evolution is time-consuming in practice and usually converges to a local minimum. The velocity of the boundary evolution and the size of the evolution step, also known as the descent direction and the step size in optimization theory, have a dramatic influence on the convergence properties. This paper focuses on developing a more efficient algorithm with faster convergence and improved performance such as smaller pattern error, lower mean edge placement error, wider defocus band, and higher normalized image log slope. These improvements are accomplished by employing the conjugate gradient of the cost function as the evolution velocity, and by introducing an optimal time step for each iteration of the boundary evolution. The latter is obtained from an extended Euler time range by using a line search method. The authors present simulations demonstrating the efficacy of these two improvements. © 2013 American Vacuum Society. [http://dx.doi.org/10.1116/1.4813781]

I. INTRODUCTION

With ever-shrinking feature size, the physical characteristics of optics have stronger impact on the imaging system. In particular, the band-limit system causes the output pattern to be a warped version of the input mask.[1,2] Several resolution enhancement techniques (RETs) have been developed to improve the performance of optical lithography.[2–4] Benefiting from the developments in computational power and optimization algorithms, inverse lithography technology (ILT), as one of these RETs, has shown great promise in meeting various challenges in future technology nodes.

Inverse lithography technology treats the mask design as an inverse mathematical problem that aims at synthesizing an input mask to deliver a desired output pattern. Various algorithms have been proposed to deal with this inverse problem in the literature. Liu and Zakhor pioneered a mask design method based on the branch-and-bound algorithm,[5–7] but this is a time-consuming process. In order to reduce the computation complexity, iterative methods were proposed to solve the inverse problem by using an optimization process.[8,9] Granik classified and discussed methods for solving inverse lithography problems in linear, quadratic, and non-linear formulations.[10] In subsequent years, Poonawala and Milanfar designed the model-based optical proximity correction (OPC) system and introduced the steepest descent (SD) algorithm for the optimization framework.[11–13] Ma and Arce

further generalized this algorithm for phase shifting masks and partially coherent imaging systems,[14–18] and proposed rigorous lithography imaging models, which take into account both 3D mask diffraction effects and optics polarization effects.[19–21] Meanwhile, the optimization algorithm was improved with an active set method by Chan *et al.*,[22] with a hotspot- and robustness-aware method using a weighted scheme by Li *et al.*,[23] combined with an augmented Lagrangian method by Li *et al.*[24] Moreover, the robustness of inverse imaging is improved through the use of stochastic gradient descent.[25–28] These methods consider the mask as a raster image constituted by pixels, where it is synthesized pixel-by-pixel. Such an approach is very flexible due to the thousands and even millions of design variables, i.e., the pixels, but such flexibility often results in patterns of much higher complexity, which in turn leads to difficulties in practical manufacturing of the input mask.[29,30]

Recently, the level set approach[31,32] has been actively explored as a feasible alternative to tackle the inverse lithography problem,[33,34] and Shen *et al.*[35] provided a complete but conventional level set formulation of this problem. In this method, the boundary of a mask is iteratively evolved according to an optimization algorithm. Such an approach is quite different from the pixel-based approach, since it does not discretize the mask into pixels, but instead it treats the mask as a continuum. Experimental results in Ref. 35 demonstrated that the level set approach achieved a considerably better performance in reducing pattern error than the pixel-based method with an almost equal runtime. Furthermore,

---

[a] Electronic mail: shyliu@mail.hust.edu.cn

Shen *et al.* pointed out that the level-set-based approach tended to eliminate small unwanted block objects in the synthesized masks, which would be generated by the pixel-based methods.[35] In addition, they took defocus and aberrations into account to enhance the robustness of the layout patterns.[36] Nevertheless, the optimization algorithm employed in the above-mentioned studies is still time-consuming, and sometimes it cannot reach an optimal solution, which limits the application in practice.

Generally, the inverse lithography problem is ill-posed, and therefore, finding a rigorous global minimum is usually too expensive and is not necessary.[7,13] Any good local minimum of the inverse lithography problem, whose goodness is evaluated using pattern error, defocus window, or other user-desired metrics, can serve as an acceptable solution. Nevertheless, various techniques are applied in ILT to try to convert it to a well-posed problem and to increase the probability of finding a better local minimum. This is accomplished by adding regularization terms,[12,13] by setting the initial mask as shown in Ref. 13, and by designing a composite cost function combining image fidelity, contour fidelity, and so on.[10,30] Yu and Yu investigated the impact of various cost functions on inverse lithography, and all these cost functions showed unique characteristics in the resulting mask patterns, optical images, and contours.[30] They found that a clever mix of these cost functions can push the resolution limits and achieve a good mask correction. In this paper, our main focus is on the impacts of iterative optimization algorithms on the solution to the inverse lithography problem. Although the iterative optimization algorithms, i.e., the evolution velocity and the time step, have seldom gained attention, they also have significant impacts on the final solution to this inverse problem, especially for the large nonlinear ill-posed lithographic system. This is because a better method can be used to search a wider solution space, and thus lead to a better solution of the inverse problem. Although Refs. 35 and 36 proposed the formulation of level-set-based inverse lithography, neither of them discussed the computational efficiency of iterative optimization algorithms, and they only mentioned that a primary SD method, which has been proved to converge relatively slowly,[37] was applied to solve this problem.

In this paper, we propose two improvements to enhance the lithographic quality and computational efficiency of the conventional level-set-based method for inverse mask synthesis. One improvement is to use the conjugate gradient (CG) method for the optimization. Among various optimization methods, the SD method, CG method, and Newton method are most often used. It has been proven that the SD method has linear convergence characteristics with respect to the iteration number, and the Newton method achieves a local quadratic rate of convergence, while the CG method attains a linear or super linear convergence rate. However, the Newton method needs too much storage to perform matrix factorizations of the Hessian matrix, i.e., the second-order derivative, and it is usually sensitive to the initial input.[37] The CG method only needs the evolution velocity of the previous iteration and therefore requires little matrix

storage compared to the Newton method.[37] It is also more efficient than the SD method, which is why it is widely used in practice to solve large linear systems of equations and non-linear optimization problems.[37] In order to achieve better performance in convergence, efficiency, and implementation, we have chosen the CG method for optimization.

The other improvement is to introduce an optimal time step during optimization. Instead of using a small constant time step for all iterations, we employ a line search method to compute the size of the time step for the boundary evolution during each iteration, and this optimal time step is capable of improving the efficiency of iteration and of achieving a better solution. In the level set method, the evolution time step during the entire optimization process is usually set to be a small constant given by the Courant–Friedrichs–Lewy (CFL) condition of the Hamilton–Jacobi equation,[32] but this method has the effect of considerably slowing down the convergence rate.

Also, most recently, we developed a fast imaging simulation algorithm for partially coherent systems by decomposing the transmission cross coefficient (TCC) into analytical kernels.[38] The large TCC matrix is projected onto the circle-sampling function (CSF) space and converted into a much smaller projected matrix. Singular value decomposition (SVD) is then performed to generate the analytical optical kernels with its eigenvectors and the CSFs. We have demonstrated that the new algorithm avoids directly performing SVD to the large TCC matrix; thus, it is much more efficient than the conventional method. In addition, the optical kernels derived by the new algorithm have analytical forms; as a result, the grid size of the kernels can be set to any desired value, which is not realizable with the conventional method. In this paper, we will apply the new algorithm to the forward lithographic modeling in order to take advantage of these improvements.

The remainder of this paper is organized as follows. Section II details the improved level set formulation for mask synthesis using the conjugate gradient and an optimal time step. Section III provides the simulation results to demonstrate the efficacy of the two proposed improvements. Finally, we draw some conclusions in Sec. IV.

II. THEORY

A. Forward lithography model and inverse problem

Abstractly, the imaging process for optical lithography is mathematically described as

$$Z(\mathbf{r}) = \Gamma\{M(\mathbf{r})\}, \tag{1}$$

where $\mathbf{r}$ represents spatial coordinates (x, y), and the operator $\Gamma\{\cdot\}$ implements the forward mapping from the input mask $M(\mathbf{r})$ to the output pattern $Z(\mathbf{r})$. In practice, $\Gamma\{\cdot\}$ in Eq. (1) consists of the projection optics effect and the resist effect.[39]

The projection optics effect, namely the optical image in resist $I(\mathbf{r})$, can be modeled as a pupil function with a partially coherent illumination source. According to Hopkins'

imaging theory,[40] the image formation process can be expressed by a bilinear transform in the spatial domain

$$I(\mathbf{r}) = \int\int TCC(\mathbf{r} - \mathbf{r}_1; \mathbf{r} - \mathbf{r}_2)M(\mathbf{r}_1)M^\dagger(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2, \quad (2)$$

where † denotes complex conjugation. The TCC is introduced as

$$TCC(\mathbf{r}_1; \mathbf{r}_2) = J(\mathbf{r}_1 - \mathbf{r}_2)H(\mathbf{r}_1)H^\dagger(\mathbf{r}_2). \quad (3)$$

Here, $J(\mathbf{r}_1 - \mathbf{r}_2)$ is the mutual intensity that describes the coherence of the illumination source; $H(\mathbf{r})$ is the point spread function that fully describes the property of the projection lens. Since TCC is Hermitian and band-limited, it possible to approximate the 4D TCC as a finite CSF series

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

where N is the total number of CSFs, $\varphi_k(\mathbf{r})$ is the kth CSF, and $p_{k,l}$ are the expansion coefficients. Since the projected matrix $\mathbf{P}$, i.e., $[p_{k,l}]$, is also Hermitian and positive definite, we can perform SVD to further decompose it 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 ith 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 4D TCC as

$$TCC(\mathbf{r}_1; \mathbf{r}_2) = \sum_{i=1}^{Q} \mu_i h_i(\mathbf{r}_1)h_i^\dagger(\mathbf{r}_2). \quad (6)$$

Here, we call $h_i(\mathbf{r})$ the ith analytical TCC kernel, which can be calculated as

$$h_i(\mathbf{r}) = \sum_{l=1}^{N} s_{i,l}\varphi_l(\mathbf{r}), \quad (7)$$

where $s_{i,l}$ is the lth element of $\mathbf{s}_i$. The flowchart of TCC decomposition into analytical kernels is shown in Fig. 1. Substituting Eq. (6) into Eq. (2), the optical image is given by

$$I(\mathbf{r}) = \sum_{i=1}^{Q} \mu_i |h_i(\mathbf{r}) \otimes M(\mathbf{r})|^2, \quad (8)$$

where $\otimes$ denotes the two-dimensional convolution.

   The resist effect can be approximated by a constant threshold resist model using the following logarithmic Sigmoid function:[12]

$$\text{sig}[I(\mathbf{r})] = \frac{1}{1 + e^{-a(I(\mathbf{r}) - t_r)}}, \quad (9)$$

where a is the steepness of the Sigmoid function, and t_r is the threshold. In reality, t_r is equal to the threshold level of the resist.

   Putting Eqs. (8) and (9) together, we rewrite Eq. (1) as

$$Z(\mathbf{r}) = \Gamma\{M(\mathbf{r})\} = \text{sig}[I(\mathbf{r})]$$
$$= \text{sig}\left[\sum_{i=1}^{Q} \mu_i |h_i(\mathbf{r}) \otimes M(\mathbf{r})|^2\right]. \quad (10)$$

   Due to the low-pass nature of the optical imaging system, $Z(\mathbf{r})$ is typically a blurred version of $M(\mathbf{r})$. We employ the L_2 norm as the pattern error to evaluate the difference between the output pattern of $M(\mathbf{r})$ and the desired pattern $Z^*(\mathbf{r})$:

$$F\{M(\mathbf{r})\} = \|\Gamma\{M(\mathbf{r})\} - Z^*(\mathbf{r})\|_2^2$$
$$= \left\|\text{sig}\left[\sum_{i=1}^{Q} \mu_i |h_i(\mathbf{r}) \otimes M(\mathbf{r})|^2\right] - Z^*(\mathbf{r})\right\|_2^2. \quad (11)$$

Now, we formulate the inverse lithography problem by finding an optimal input mask $M^*(\mathbf{r})$ to minimize the pattern error, i.e., the cost function $F\{M(\mathbf{r})\}$,

$$M^*(\mathbf{r}) = \underset{M(\mathbf{r})}{\arg\min}\, F\{M(\mathbf{r})\}. \quad (12)$$
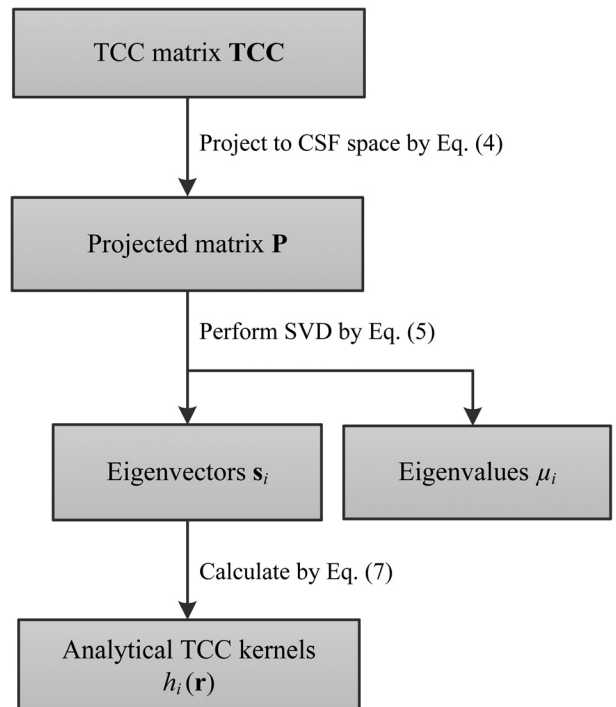

FIG. 1. Flowchart of TCC decomposition into analytical kernels.

B. Preliminaries

Before introducing the level set formulation of the optimization problem, the CG method, and the optimal time step, we will first explain some of the terms which are commonly encountered in this work.

1. Polak–Ribière–Polyak conjugate gradient method

In the CG method, the evolution velocity v_k in the kth iteration is defined by

$$v_k = \begin{cases} -g_k + \eta_k v_{k-1} & \text{if } k \geq 1 \\ -g_k & \text{if } k = 0 \end{cases}.$$ (13)

Here, g_k is a matrix representing the gradient $\nabla F\{M(\mathbf{r}, t_k)\}$ of $F\{M(\mathbf{r}, t_k)\}$ at $\mathbf{r}$, 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 is simplified as the SD method. In the widely used Polak–Ribière–Polyak (PRP) CG method, η_k is defined as[41,42]

$$\eta_k^{\text{PRP}} = \frac{\|g_k\|_2^2 - \sum g_k \cdot g_{k-1}}{\|g_{k-1}\|_2^2}.$$ (14)

In practical computations, the PRP CG method is generally believed to be the most efficient CG method.[42] Compared to the SD method, the PRP CG method 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.[42] In a sense, it can search a wider solution space and has the capability of achieving a better solution. For theoretical foundations and demonstrations, interested readers may refer to Refs. 37, 41, and 42.

2. Courant–Friedrichs–Lewy condition and the Euler time step

The CFL condition is necessary for convergence and stability when solving certain partial differential equations. It asserts that the numerical wave speed $\Delta x / \Delta t$ must be at least as fast as the physical wave speed $|u|$, i.e., $\Delta x / \Delta t > |u|$ (see Chap. 3.2 in Ref. 32). In the present application, the CFL condition is given by

$$\Delta t \cdot \max\left\{ \frac{|v_x|}{\Delta x} + \frac{|v_y|}{\Delta y} \right\} = \alpha,$$ (15)

where Δx and Δy provide the grid size of the discrete Cartesian grid, v_x and v_y are the components of evolution velocity v in the x and y direction, respectively. α is called the CFL number with $0 < \alpha < 1$, and the calculated time step Δt based on Eq. (15) is called the Euler time step.

3. Weighted essentially nonoscillatory polynomial

It is possible to evaluate the spatial derivatives of level set function ϕ by using a first-order accurate forward difference and backward difference, or a second-order accurate central difference.[43] In order to improve the computational accuracy, Harten *et al.* developed the essentially nonoscillatory (ENO) polynomial interpolation method that uses the smoothest possible polynomial interpolation of ϕ, and then differentiates it to get the spatial derivatives.[44] Frequently used ENO schemes include first-, second-, and third-order accurate essentially nonoscillatory (ENO1, ENO2, ENO3) methods, in addition to a fifth-order accurate weighted ENO (WENO) method, which consists of a convex combination of ENO approximations. In this work, we use the WENO method to get the level set function ϕ. Interested readers may refer to Refs. 32, 43, and 44 for more details.

C. Level set formulation with conjugate gradient and optimal time step

In the level-set based framework of inverse lithography, the $M(\mathbf{r})$ is described by a level set function $\phi(\mathbf{r})$, which is related to $M(\mathbf{r})$ by

$$M(\mathbf{r}) = \begin{cases} m_{\text{int}} & for\{\mathbf{r} : \phi(\mathbf{r}) < 0\} \\ m_{\text{ext}} & for\{\mathbf{r} : \phi(\mathbf{r}) > 0\} \end{cases},$$ (16)

where $m_{\text{int}} = 1$ and $m_{\text{ext}} = 0$ when a binary mask is considered. The boundary of subregions in $M(\mathbf{r})$ is the zero level of $\phi(\mathbf{r})$, namely, $\phi(\mathbf{r}) = 0$.

Generally, a distance function $d(\mathbf{r})$ is defined as

$$d(\mathbf{r}) = \min(|\mathbf{r} - \mathbf{r}_\Omega|), for\ all \quad \mathbf{r}_\Omega \in M_\Omega,$$ (17)

where M_Ω denotes the boundaries of $M(\mathbf{r})$, and $\phi(\mathbf{r})$ is defined as the signed distance function[32] related to $d(\mathbf{r})$

$$\phi(\mathbf{r}) = \begin{cases} -d(\mathbf{r}) & \mathbf{r} \in M_{\Omega-} \\ 0 & \mathbf{r} \in M_\Omega \\ +d(\mathbf{r}) & \mathbf{r} \in M_{\Omega+} \end{cases},$$ (18)

where $M_{\Omega-}$ is the region occupied by the mask and $M_{\Omega+}$ is the void region.

Now, we reformulate the inverse lithography problem by finding an optimal level set function $\phi^*(\mathbf{r})$ that seeks to minimize the cost function $F\{M(\mathbf{r})\}$ as

Finding $\phi^*(\mathbf{r})$ to minimize $F\{M(\mathbf{r})\}$,

$$\text{where}: M(\mathbf{r}) = \begin{cases} m_{\text{int}} & for\ \{\mathbf{r} : \phi^*(\mathbf{r}) < 0\} \\ m_{\text{ext}} & for\ \{\mathbf{r} : \phi^*(\mathbf{r}) > 0\} \end{cases}.$$

In this way, the inverse lithography problem is changed to handle the level set function $\phi(\mathbf{r})$ instead of the mask pattern $M(\mathbf{r})$ itself. To solve this inverse problem, an iterative scheme is applied by deriving an evolution equation for the level set function $\phi(\mathbf{r})$. Letting t represent the artificial time, we arrive at the Hamilton–Jacobi equation

$$\frac{\partial \phi}{\partial t} = v(\mathbf{r}, t) |\nabla \phi|,$$ (19)

where $v(\mathbf{r}, t)$ is the evolution velocity of $\phi(\mathbf{r})$ at $\mathbf{r}$ in the normal direction. In the optimization process, the evolution

velocity $v(\mathbf{r}, t)$ should reduce the cost function F{M}, namely δF{M} < 0, where δF{M} is the directional derivative of the cost function F{M} in the direction δM. According to Ref. 45, the directional derivative δF{M} is

$$\delta\mathrm{F}\{M\} = \sum \nabla\mathrm{F}\{M\} \cdot v(\mathbf{r}, t), \qquad (20)$$

where $\cdot$ denotes an element-by-element multiplication operator, and ∇F{M} is the gradient with respect to M

$$\begin{aligned}
\nabla \mathrm{F}\{M\} &= J(M)^{\mathrm{T}}[\Gamma\{M\} - Z^*] \\
&= a\left\{\sum_{i=1}^{Q} \lambda_i h_i^{flip} \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i^\dagger \otimes M)]\right\} \\
&\quad + a\left\{\sum_{i=1}^{Q} (\lambda_i h_i^{flip})^\dagger \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i \otimes M)]\right\}.
\end{aligned}$$
$$(21)$$

The detailed derivation of Eqs. (20) and (21) are given in the Appendix. Here, $J(M)$ is the Jacobian of $\Gamma\{M\}$ at M; h_i^{flip} is the up–down and left–right flip of h_i; and a is the steepness of the Sigmoid function in Eq. (9).

Generally, the evolution velocity $v(\mathbf{r}, t)$ is set to be the negative gradient of the cost function F{M},[35,37] i.e., $v(\mathbf{r}, t) = -\nabla\mathrm{F}\{M\}$, which leads to $\delta\mathrm{F}\{M\} = -\sum |\mathrm{F}\{M\}|^2 < 0$ and implies the descent of F{M}. In the present work, the conjugate gradient of F{M} is used for the evolution velocity $v(\mathbf{r}, t)$, which also satisfies a negative δF{M}.

Now, the procedure for the level set method with conjugate gradient and optimal time step for inverse mask synthesis is described as follows:

Iteration 0: Given a desired output pattern $Z^*(\mathbf{r})$, we use $Z^*(\mathbf{r})$ as the initial input mask $M(\mathbf{r}, t_0)$, then compute the level set function $\phi(\mathbf{r}, t_0)$ of $M(\mathbf{r}, t_0)$, and calculate the gradient g_0 of F{$M(\mathbf{r}, t_0)$} with respect to $M(\mathbf{r}, t_0)$, which is based on Eq. (21) and the evolution velocity $v(\mathbf{r}, t_0)$, which is based on Eq. (13).
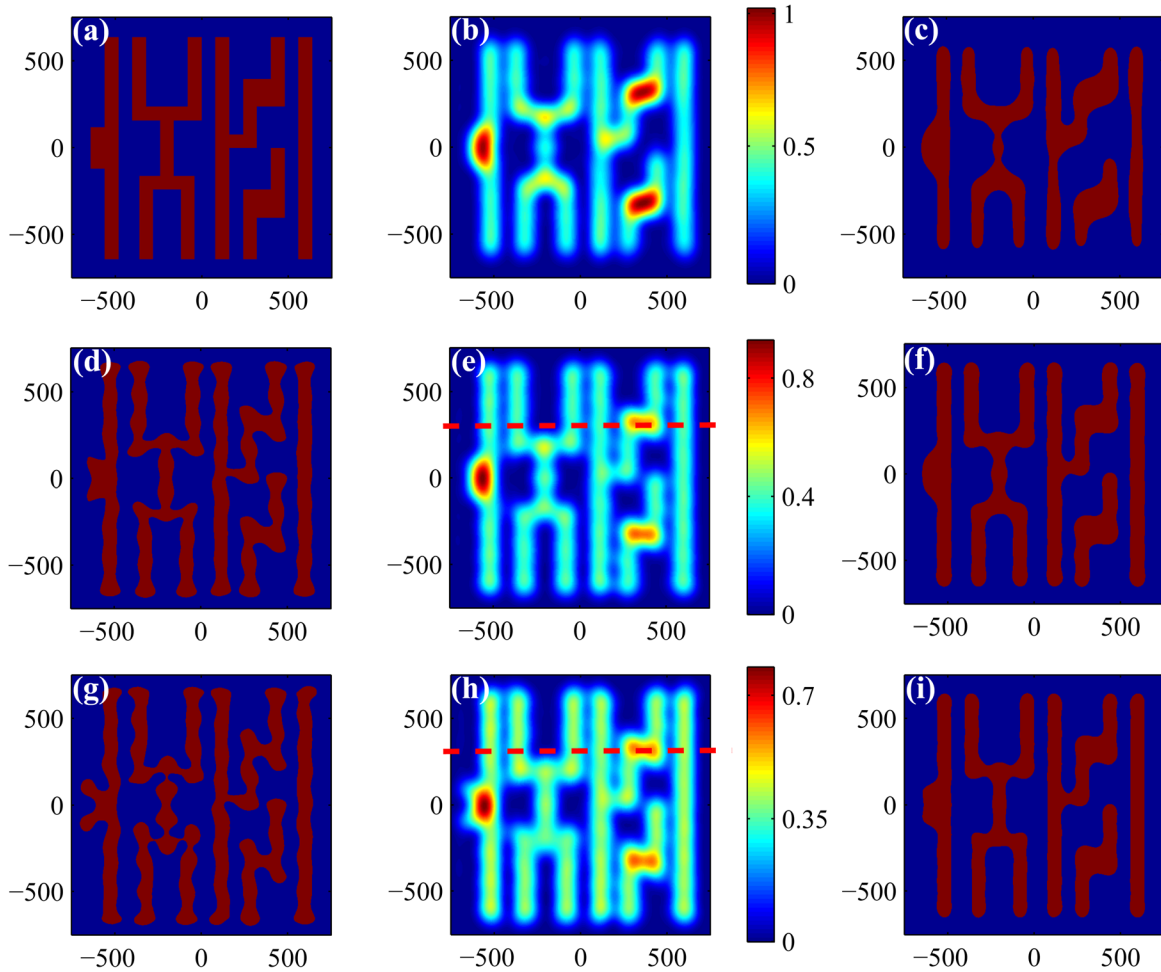

FIG. 2. (Color online) (a) Desired pattern, (b) its optical image, and (c) its output pattern on the wafer with a pattern error of 16 723. (d) The synthesized mask pattern by the SD method after 81 iterations, (e) its optical image, and (f) its output pattern on the wafer with a pattern error of 4408. (g) The synthesized mask pattern by the CG method after 81 iterations, (h) its optical image, and (i) its output pattern on the wafer with a pattern error of 3670. The horizontal axis and vertical axis denote x position and y position of the patterns in nanometers, respectively. The two red dotted lines are CD cutlines both at the position of $y = 450$ nm, which will be detailed in Fig. 5(b).

$$\phi(\mathbf{r}, t_0) = \begin{cases} -d(\mathbf{r}) & \mathbf{r} \in M(\mathbf{r}, t_0)_{\Omega-} \\ 0 & \mathbf{r} \in M(\mathbf{r}, t_0)_{\Omega} \\ +d(\mathbf{r}) & \mathbf{r} \in M(\mathbf{r}, t_0)_{\Omega+} \end{cases}, \quad (22)$$

$$g_0 = \nabla F\{M(\mathbf{r}, t_0)\}, \quad (23)$$

$$v(\mathbf{r}, t_0) = -g_0. \quad (24)$$

Iteration k:

Step i: Calculate the Euler time step Δt_k^e according to Eq. (15), and then employ a line search method to obtain an optimal time step Δt_k from an extended Euler time range

$$\Delta t_k = \arg \min_{\Delta t} F\{M(\mathbf{r}, t_k + \Delta t)\}. \quad (25)$$

Here Δt is a time step within the search range $[\beta \cdot \Delta t_k^e, \gamma \cdot \Delta t_k^e]$ specified by the Euler time step Δt_k^e, where β and γ are called the extension factors to control the search range, and $0 \leq \beta < \gamma$.

Step ii: Update $\phi(\mathbf{r}, t_{k+1})$ and $M(\mathbf{r}, t_{k+1})$,

$$\phi(\mathbf{r}, t_{k+1}) = \phi(\mathbf{r}, t_k) + v(\mathbf{r}, t_k) \Delta t_k, \quad (26)$$

$$M(\mathbf{r}, t_{k+1}) = \begin{cases} m_{\text{int}} & for\{\mathbf{r} : \phi(\mathbf{r}, t_{k+1}) < 0\} \\ m_{\text{ext}} & for\{\mathbf{r} : \phi(\mathbf{r}, t_{k+1}) > 0\} \end{cases}. \quad (27)$$

Step iii: Calculate the gradient g_{k+1} based on Eq. (21), the factor η_{k+1} based on Eq. (14), and then the evolution velocity $v(\mathbf{r}, t_{k+1})$ for the next iteration

$$g_{k+1} = \nabla F\{M(\mathbf{r}, t_{k+1})\}, \quad (28)$$

$$\eta_{k+1}^{\text{PRP}} = \frac{\|g_{k+1}\|_2^2 - \sum g_{k+1} \cdot g_k}{\|g_k\|_2^2}, \quad (29)$$

$$v(\mathbf{r}, t_{k+1}) = -g_{k+1} + \eta_k^{\text{PRP}} \cdot v(\mathbf{r}, t_k). \quad (30)$$

If $\|v(\mathbf{r}, t_{k+1})\| < \zeta$ or $k > \Psi$, go to **Stop.**
Else, return to **Step i**.
Stop: Obtain the synthesized mask,

$$M^*(\mathbf{r}) = M(\mathbf{r}, t_{k+1}). \quad (31)$$

In the above procedure, the iteration is terminated when $\|v(\mathbf{r}, t_{k+1})\| < \zeta$ or $k > \Psi$, where ζ is defined as the minimum value of the norm of velocity, and Ψ is the prescribed upper
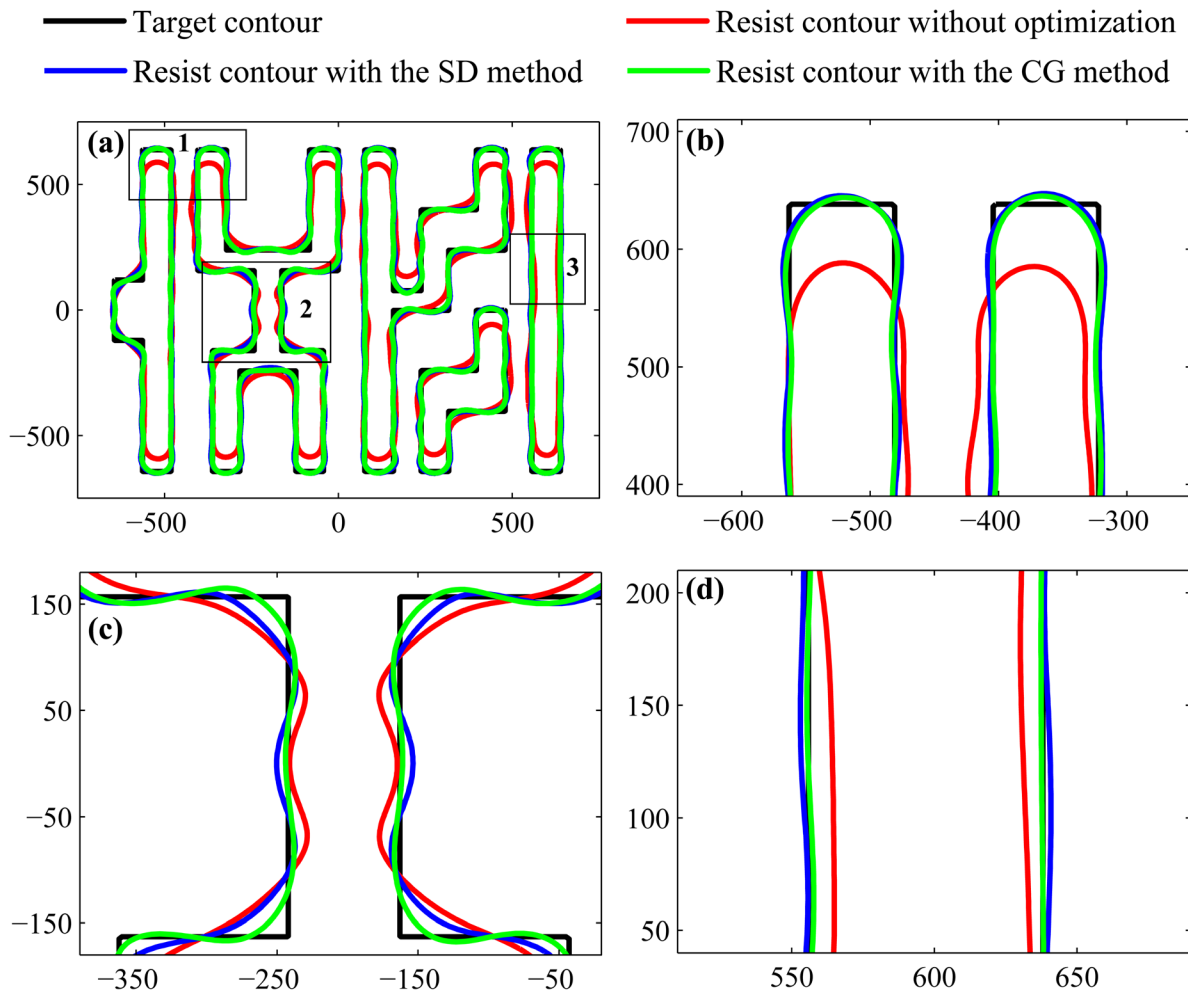

FIG. 3. (Color online) Comparison of output pattern contours. Enlarged views of three black boxes 1, 2, and 3 in (a) are shown in (b), (c), and (d) for the line-end error, the corner error, and the edge error, respectively.

limit of the number of iterations. The termination criterion $\|v(\mathbf{r}, t_{k+1})\| < \zeta$ means that the evolution stops when the velocity along the boundary is zero or rather small.

A small constant time step, such as the Euler time step, is commonly used for optimization, and this is a very safe method for enforcing the convergence stability of the evolution. However, it likely results in a large number of iterations. Hence, we introduce an optimal time step to improve the efficiency of iteration and provide the capability of achieving a better solution. In a strict sense, the optimal time step is not the exact optimal solution, which would be too expensive to compute. In this specific inverse lithography problem, when the time step is too small or too large, the iteration gave rise to bad performance during our simulation experiments. This result led us to assume that the problem of seeking the optimal time step was locally convex, although without theoretical foundations. Then, the optimal time step was obtained using a line search method from a given search range, which was specified as an extended Euler time range $[\beta \cdot \Delta t_k^e, \gamma \cdot \Delta t_k^e]$ by simply setting the extension factors

β and γ. The optimal time step shows a much faster convergence rate over that achieved by the constant time step, which has been demonstrated in our simulations.

D. Regularization

Since inverse lithography is generally ill-posed, regularization approaches are applied to bias the mask to a more desired pattern. In this work, we adopt total variation (TV) regularization to reduce mask complexity,[12,13,35] which is defined as[46]

$$R(M) = \int_{\Omega} |\nabla M| d\mathbf{r}, \tag{32}$$

where Ω is the area of the bounded domain, or the number of pixels in M. TV regularization normally leads to a curve shortening result and makes the boundary move in the normal direction with a velocity defined by

$$\sigma(\mathbf{r}, t) = \nabla \cdot \left(\frac{\nabla M}{|\nabla M|} \right). \tag{33}$$

By adding the regularization velocity as shown in Eq. (33), we update Eq. (19) as
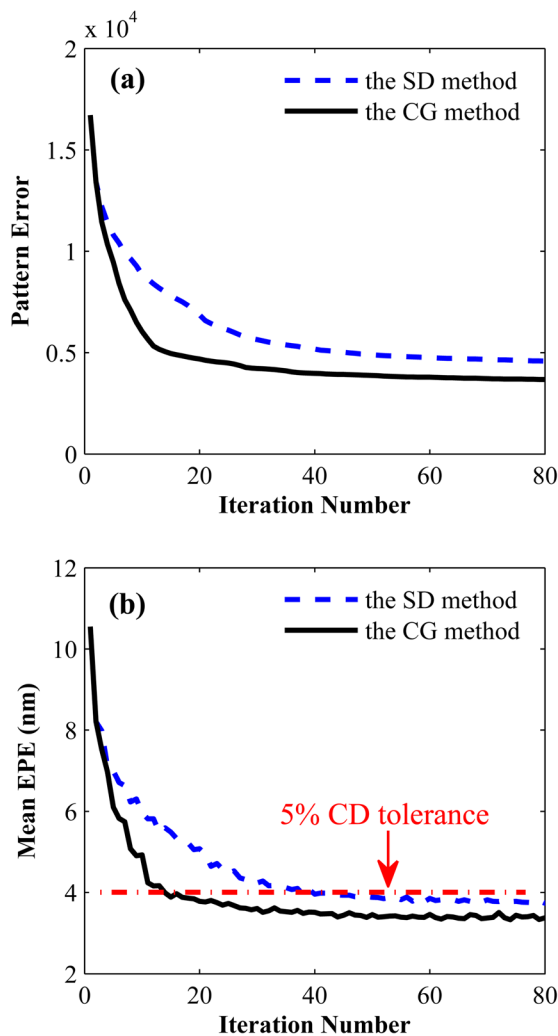

FIG. 4. (Color online) Convergence of pattern error and mean EPE by the SD and CG methods, both for the desired pattern as shown in Fig. 2(a). The red dotted–dashed line represents the 5% CD tolerance.
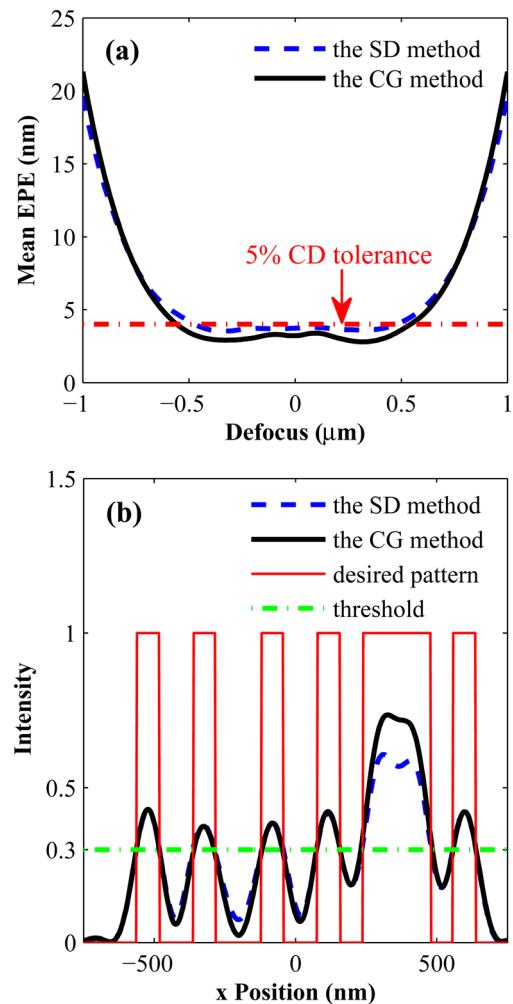
FIG. 5. (Color online) (a) Mean EPEs vs defocus variation for the synthesized mask pattern shown in Figs. 2(d) and 2(g), and (b) intensity distributions along the cutlines shown in Figs. 2(e) and 2(h).

$$\frac{\partial \phi}{\partial t} = v(\mathbf{r}, t)|\nabla \phi| + \kappa \cdot \sigma(\mathbf{r}, t)|\nabla \phi|, \qquad (34)$$

where κ is a user-defined parameter to reveal the weight of the TV regularization.

III. SIMULATION

Simulations were performed on a partially coherent imaging system with an annular source illumination whose outer radius was $\sigma_{out} = 0.6$ and whose inner radius was $\sigma_{in} = 0.4$. The wavelength in the simulations was set at 193 nm, and the numerical aperture was 0.85. The resist effects were approximated by a Sigmoid function with $a = 80$ and $t_r = 0.3$. The TV regularization factor κ was set at 0.01. The value of ζ was set at 0.3 in the termination criterion $\|v(\mathbf{r}, t_{k+1})\| < \zeta$, where the norm of the evolution velocity $\|v(\mathbf{r}, t_{k+1})\|$ was normalized to $\|v(\mathbf{r}, t_0)\|$. The WENO polynomial was applied to calculate the level set function $\phi(\mathbf{r})$ and its spatial derivatives. The CFL number α was set at 0.5. The Euler time step was calculated using the CFL number α and Eq. (15). We applied the Golden Section Search

method[47] as the line search method, and the search range was set at $[0.1\Delta t^e, 10\Delta t^e]$ with a terminated accuracy of $0.01\Delta t^e$, where Δt^e is the Euler time step. All the simulations were carried out with in-house MATLAB codes on a HPZ800 (3.47 GHz Xeon) Workstation using a Windows 7 (64 bit) operating system.

Figure 2 depicts the simulated images for the synthesized mask patterns by using the SD and CG methods. Figure 2(a) shows the desired pattern, which is with a critical dimension (CD) of 80 nm and consists of 501×501 pixels with a grid resolution of 3 nm. As expected, the optical images and the output patterns on the wafer differ for different input mask patterns, and the synthesized mask patterns by the SD and CG methods achieve much smaller pattern errors compared to that obtained by simply inputting the desire pattern as the mask pattern. After the same 81 iterations, the CG method results in a pattern error of 3670 compared to 4408 by the SD method, which means a performance improvement by 20.1%. Moreover, we compared the output pattern contours without mask optimization, with the SD optimization method, and with the CG optimization method. As shown in
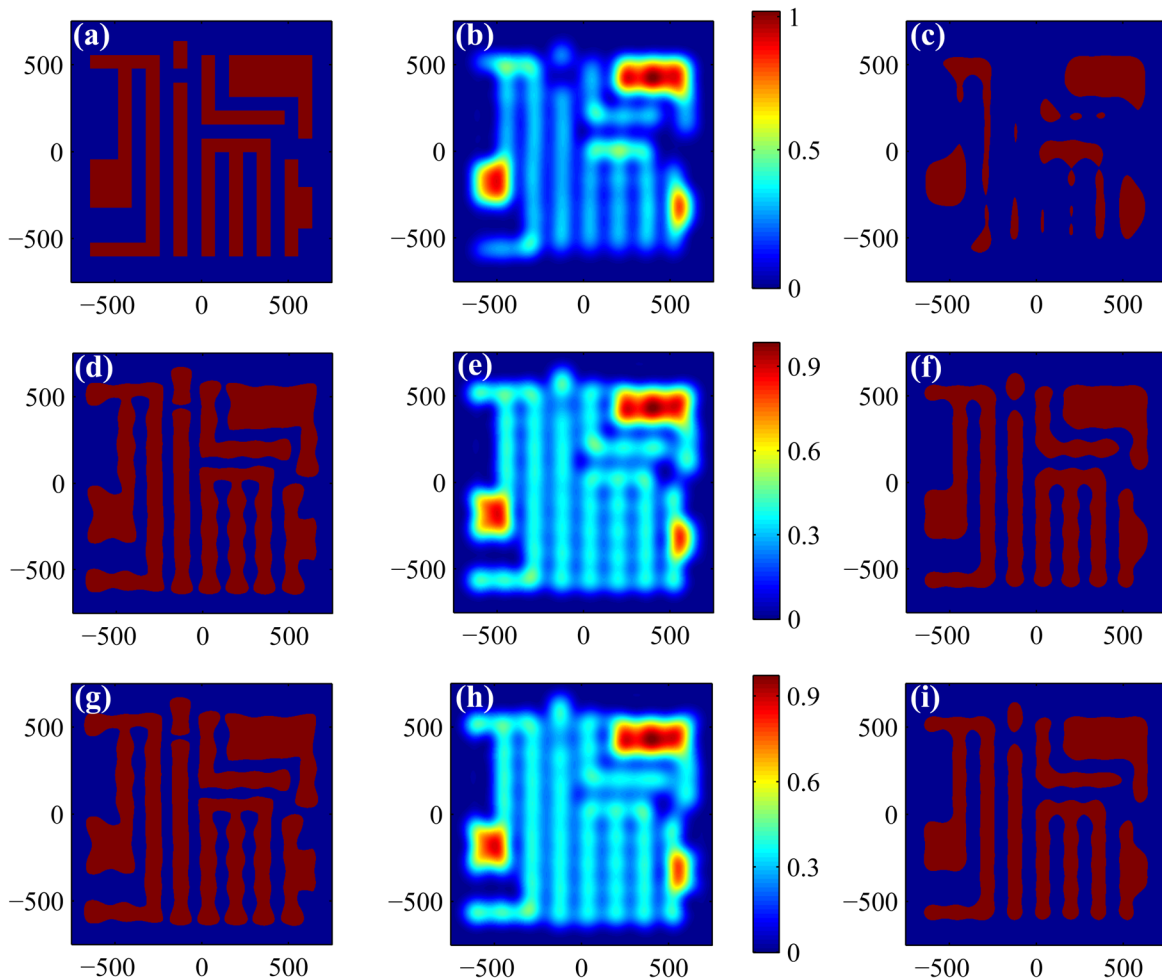

FIG. 6. (Color online) (a) Desired pattern, (b) its optical image, and (c) its output pattern on the wafer with a pattern error of 42 066. (d) The synthesized mask pattern by the SD method after 100 iterations, (e) its optical image, and (f) its output pattern on the wafer with a pattern error of 6781. (g) The synthesized mask pattern by the CG method after 100 iterations, (h) its optical image, and (i) its output pattern on the wafer with a pattern error of 5372. The horizontal and vertical axes denote x position and y position of the patterns in nanometers, respectively.[47]
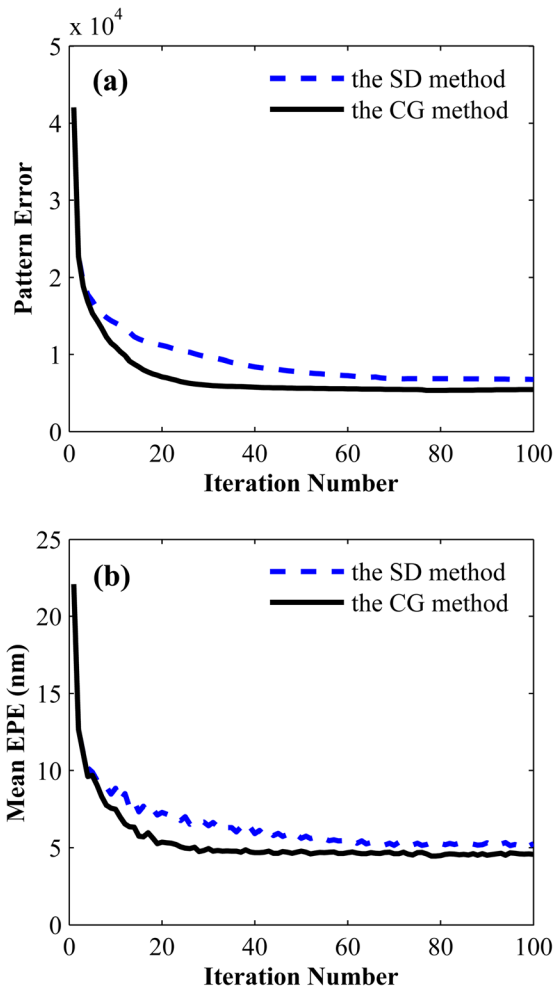
FIG. 7. (Color online) Convergence of pattern error and mean EPE by the SD and CG methods, both for the desired pattern as shown in Fig. 6(a).

Fig. 3, it is obvious that the resist contour without mask optimization is a distorted version of the input mask, while this distortion is corrected dramatically by using the SD or CG method. From the enlarged views shown in Figs. 3(b) to

3(d), it is also observed that the CG method achieves a relatively smaller contour error than that by the SD method, especially around the corners, as shown in Fig. 3(c). This is because the CG method searches a wider solution space and hence achieves a better solution than the SD method.

Edge placement error (EPE) is popularly used in RETs, especially in the polygon-based OPC, to convey CD information, which is essentially the CD error at one side.[48,49] Here, the mean EPE is also introduced to evaluate the optimization effect, which is defined as the weighted average EPE of all segments

$$\text{mean EPE} = \frac{\sum_{i=1}^{S} \omega_i |\text{EPE}(p_i)|}{\sum_{i=1}^{S} \omega_i}, \tag{35}$$

where p_i is the ith segment, and S is the total number of segments. In this work, all the weighted factors ω_i are set at 1.

Figure 4 illustrates the convergence comparison, for both the pattern error and the mean EPE. In Fig. 4(a), the SD method converges to a pattern error of 4408 with 81 iterations after 1494 s, while the CG method converges to a pattern error of 4362 with 27 iterations after 499 s. The CG method also converges to a pattern error of 3670 with 81 iterations after 1494 s. From Fig. 4(b), it is interesting to observe that the mean EPE generally has a positive relationship with the pattern error along with a small local fluctuation. This means that the mean EPE is equivalent to the pattern error. However, in practice, the mean EPE is rarely adopted as a cost function in pixel-based OPC, especially in level-set-based inverse lithography, because the calculation of the explicit gradient of the mean EPE is too complicated or even impossible. We therefore directly use the pattern error as the cost function, as shown in Eqs. (11) and (12) to guide mask synthesis in this work, but we adopted both the pattern error and the mean EPE as metrics to evaluate the

TABLE I. Optimization comparison for some typical mask patterns by the SD and CG methods, both using an optimal time step.[a]

Mask patterns	Iteration number to fulfill 10% CD tolerance		Iteration number to fulfill 5% CD tolerance		Reduction of pattern error by the CG method compared to the SD method (%)
	The SD method	The CG method	The SD method	The CG method	
Single contact hole	1	1	1	1	4.7
Single bar	1	1	2	2	2.4
Single crossbar	2	2	12	7	12.2
Nine contact holes (sparse)[b]	6	4	–[c]	16	26.7
Nine contact holes (dense)	8	6	–	–	11.7
Seven bars (sparse)	2	2	2	2	12.0
Seven bars (dense)	7	3	16	9	21.5
Figure 2(a)	3	3	40	13	20.1
Figure 6(a)	16	8	–	–	26.2

[a]All the target CD of the mask patterns is 80 nm.
[b]Sparse denotes the ratio of CD to pitch is 1:3; Dense denotes the ratio of CD to pitch is 1:2.
[c]– means that it cannot satisfy the condition.

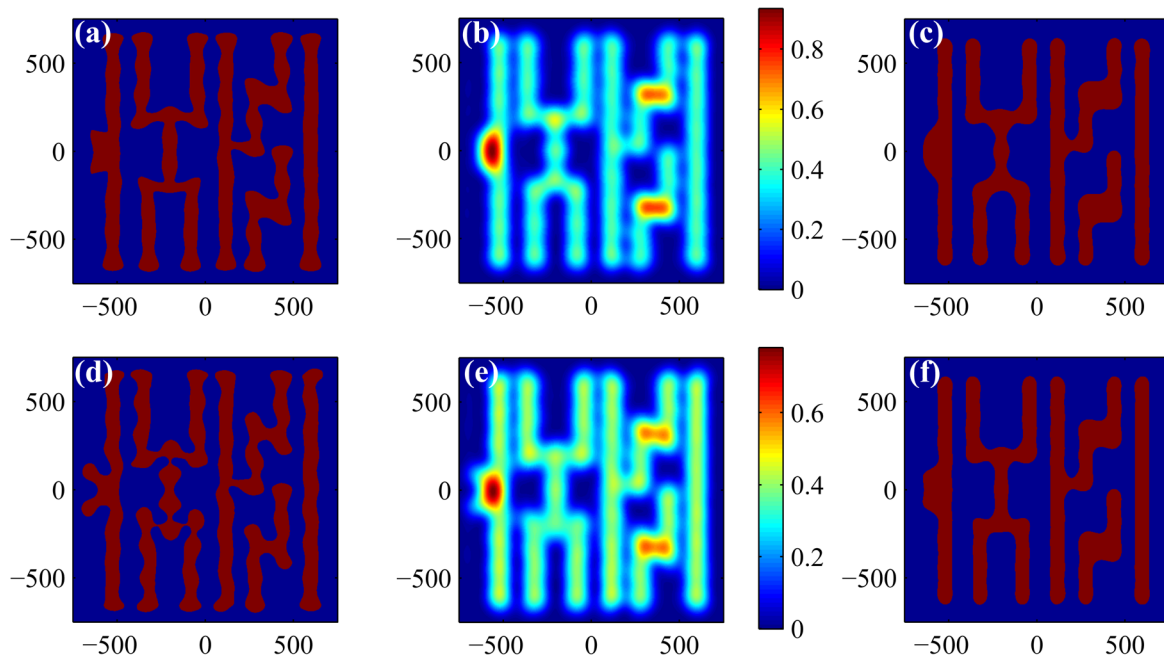**JVST B - Microelectronics and Nanometer Structures**

Fig. 8. (Color online) (a) Synthesized mask pattern by the CG method with the Euler time step after 151 iterations, (b) its optical image, and (c) its output pattern on the wafer with a pattern error of 4785. (d) The synthesized mask pattern by the CG method with the optimal time step after 81 iteration, (e) its optical image, and (f) its output pattern on the wafer with a pattern error of 3670. The horizontal axis and vertical axis denote x position and y position of the patterns in nanometers, respectively.

optimization effect. As shown in Fig. 4(b), both the SD and CG methods obtain a mask pattern which can fall within a 5% tolerance of the CD margin, or 4 nm in this case (the target CD is 80 nm); the CG method, however, needs only 13 iterations to fulfill this goal, while the SD method needs 40 iterations. It is clear that the CG method shows a pronounced convergence performance with high computational efficiency.

The normalized image log slope (NILS) is usually used to evaluate the sensitivity of the mask pattern to dose variation,[50] and is given by

$$\mathrm{NILS} = \left| \frac{\mathrm{CD}}{I_{\mathrm{threshold}}} \frac{dI}{dx} \Big|_{I_{\mathrm{threshold}}} \right|, \qquad (36)$$

where $I_{\mathrm{threshold}}$ is the threshold intensity [i.e., t_r in our resist model Eq. (9)], and CD is the critical dimension (80 nm in this case). A higher value of NILS is preferred in practice.

Figure 5 depicts the mean EPE varying with defocus and the intensity distribution along a cutline. From Fig. 5(a), it is revealed that the mask pattern synthesized by the CG method achieves a wider defocus band in fulfilling a 5% CD tolerance. Compared to the mask pattern synthesized by the SD method, this mask pattern also achieves a lower mean EPE under the same defocus conditions. In Fig. 5(b), the average NILS of 12 edge slopes obtained by the CG method is 1.3425 compared to 1.2953 by the SD method, indicating that the proposed CG method also improves the tolerance to dose fluctuation. Moreover, the contrast of the optical image by the CG method is sharply contoured compared to that by the SD method, as shown in Figs. 2 and 5(b).
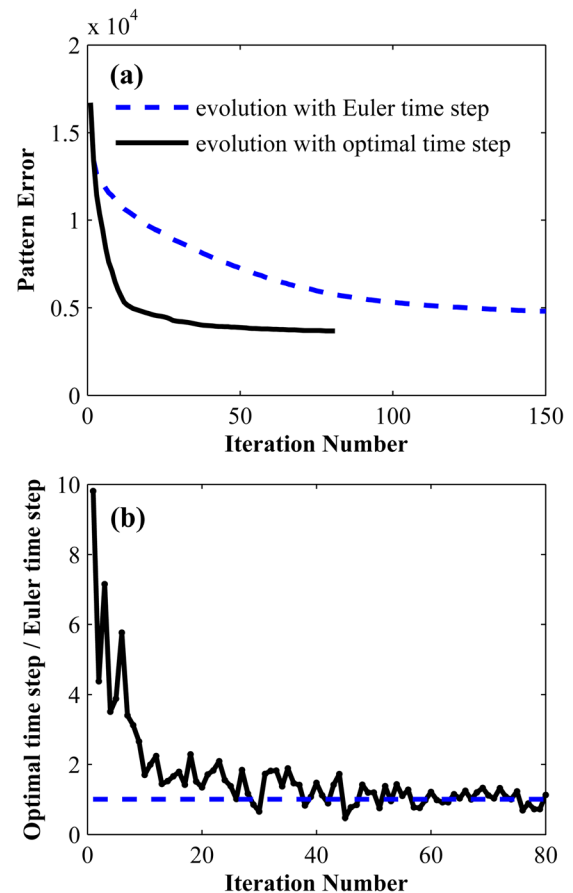

Fig. 9. (Color online) (a) Convergence of pattern error by the CG method with the optimal time step and the Euler time step, and (b) the ratio of the optimal time step to the Euler time step during the evolution, both for the desired pattern shown in Fig. 2(a).

TABLE II. Comparison of the pattern error and convergence performance by the CG method with different time steps.

Time step	Pattern error	Iteration number	Runtime (s)	Time/iteration (s)
The Euler time step	4785	151	2147	14.2
The optimal time step	3670	81	1494	18.5

Another set of simulations is presented in Figs. 6 and 7, where the desired pattern has a size of 501×501 pixels with a grid resolution of 3 nm. The optimization is terminated after 100 iterations. Compared to the results by the SD method, the CG method reduces the pattern error by 26.2%, as shown in Fig. 6. The CG method also achieves a better convergence with the same running time (or iteration number) and has a pronounced improvement in the convergence rate, as shown in Fig. 7.

It should be noted that the CG method usually achieves a slightly better mask pattern than the SD method when optimizing a simple mask pattern during our simulation experiments, like a single bar, a single contact hole, a patterns with very sparse features, etc., but the CG method always results in a considerable improvement in computational efficiency. On the other hand, when handling a more complicated mask

pattern, as shown in Figs. 2(a) and 6(a), the CG method outperforms dramatically over the SD method. From our two sets of simulations, the CG method has demonstrated the capacity to reduce the pattern error by more than 20%. Table I summarizes the optimization performance for some typical mask patterns by using the SD and CG methods.

We also performed simulations for the mask pattern shown in Fig. 2(a) by using the CG method with two different time steps. The optimization results and the convergence history are depicted in Figs. 8 and 9, respectively. As illustrated in Fig. 9(a), adopting the Euler time step achieves a pattern error of 4785 with 151 iterations after 2147 s, while evolving with the optimal time step achieves a pattern error of 3670 with 81 iterations after 1494 s. It is interesting to note that the evolution with the optimal time step converges faster than that with the Euler time step, especially in the first tens of iterations. Although the proposed method takes more time in each iteration, the total time is less than that with the Euler time step (see Table II). Figure 9(b) compares the optimal time step with the Euler time step calculated during the optimization process. Note that after certain iterations, the optimal time step size barely changes, indicating that a constant time step size can be used for the subsequent iterations.

Figure 10 depicts the calculated cost function varying with respect to the time step Δt in the extended Euler time range of $[0.1\Delta t^e, 10\Delta t^e]$, by taking the 1st and the 60th iterations as two examples. It is clear that the cost function in each iteration is approximately convex within the extended Euler time range, and therefore, a line search method such as the Golden Section Search method can easily find an optimal time step.

IV. CONCLUSIONS

This paper proposes two improvements for level-set-based inverse lithography. One improvement uses the CG method for optimization. Compared to the conventional SD method, the CG method results in improved performance with a lower mean EPE, a wider defocus band, and a higher NILS, along with a considerable improvement in computational efficiency. For the second improvement, we introduce an optimal time step to accelerate the iteration with a higher efficiency. In addition, we find that the optimal time step size obtained by a line search method barely changes after certain iterations, with the result that a constant step size can then be used for subsequent iterations to decrease computing time. The proposed method is expected to have applications in mask optimization and synthesis for optical lithography.
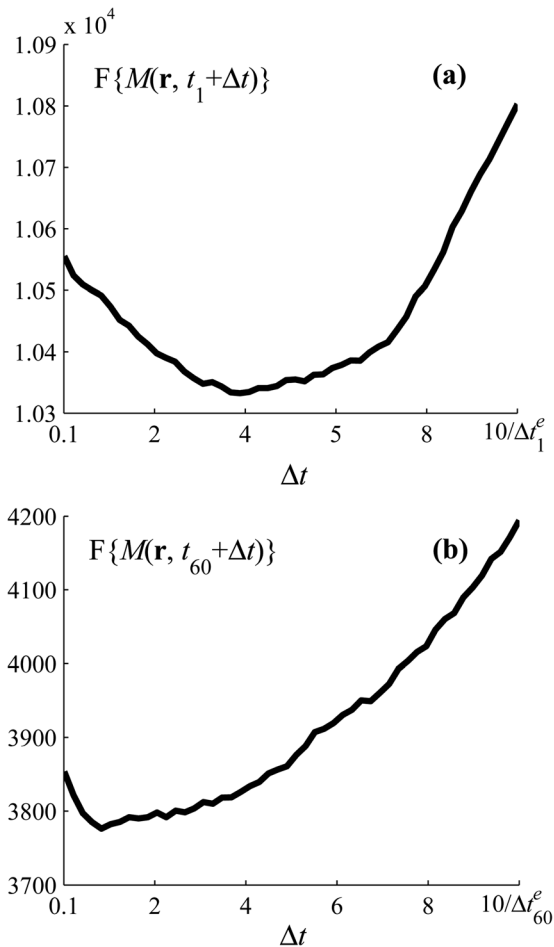

FIG. 10. Cost function vs time step Δt in the extended Euler time range of $[0.1\Delta t^e, 10\Delta t^e]$, where $F\{M(\mathbf{r}, t_1 + \Delta t)\}$ and $F\{M(\mathbf{r}, t_{60} + \Delta t)\}$ denote the cost functions in the 1st and the 60th iterations, respectively, and Δt_1^e and Δt_{60}^e representing the corresponding Euler time steps.

APPENDIX A: DERIVATION OF EQ. (20)

According Ref. 45, the unit outward normal of the boundary of M at $\mathbf{r}$ is

$$\vec{n}(\mathbf{r}) = \frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|}. \tag{A1}$$

δM and $\delta\mathbf{r}$ represent the small variations of M and $\mathbf{r}$, respectively. Assuming that each point moves perpendicular to the boundary, i.e., $\vec{n}(\mathbf{r})$, δM can be expressed as the normal displacement

$$\delta M = \delta\mathbf{r} \cdot \frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|}. \tag{A2}$$

If the normal evolution velocity is $v(\mathbf{r}, t)$, it leads to

$$\delta\mathbf{r} = v(\mathbf{r}, t)\frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|}. \tag{A3}$$

Noticing Eqs. (A1)–(A3), we can therefore derive $\delta F\{M\}$ as

$$\delta F\{M\} = \sum \nabla F\{M\} \cdot \delta M(\mathbf{r}) = \sum \nabla F\{M\} \cdot \frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|} \cdot \delta\mathbf{r} = \sum \nabla F\{M\} \cdot \frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|} \cdot v(\mathbf{r}, t) \cdot \frac{\nabla\phi(\mathbf{r})}{|\nabla\phi(\mathbf{r})|}$$
$$= \sum \nabla F\{M\} \cdot v(\mathbf{r}, t). \tag{A4}$$

APPENDIX B: DERIVATION OF EQ. (21)

To derive the gradient in Eq. (21), we first give some useful intermediate results as

$$\frac{\partial \mathrm{sig}(x)}{\partial x} = \frac{\partial \frac{1}{1+e^{-a(x-t_r)}}}{\partial x} = a \cdot \left(\frac{1}{1+e^{-a(x-t_r)}}\right)^2 \cdot \left(e^{-a(x-t_r)}\right) = a \cdot \left(\frac{1}{1+e^{-a(x-t_r)}}\right) \cdot \left(1 - \frac{1}{1+e^{-a(x-t_r)}}\right)$$
$$= a \cdot \mathrm{sig}(x) \cdot [1 - \mathrm{sig}(x)], \tag{B1}$$

and

$$\frac{\partial[h(\mathbf{r}) \otimes M(\mathbf{r})]}{\partial M(\boldsymbol{\rho})} = \frac{\partial\left[\sum_{\tau} h(\mathbf{r} - \tau)M(\tau)\right]}{\partial M(\boldsymbol{\rho})} = h(\mathbf{r} - \boldsymbol{\rho}), \tag{B2}$$

where $\mathbf{r}$, τ and $\boldsymbol{\rho}$ denote the spatial coordinates (x, y).

As $M(\mathbf{r})$ is real, we express I_i as

$$I_i = |h_i(\mathbf{r}) \otimes M(\mathbf{r})|^2 = [h_i(\mathbf{r}) \otimes M(\mathbf{r})] \cdot [h_i^{\dagger}(\mathbf{r}) \otimes M(\mathbf{r})]. \tag{B3}$$

Then, the derivation of I_i with respect to $M(\boldsymbol{\rho})$ is

$$\frac{\partial I_i}{\partial M(\boldsymbol{\rho})} = \frac{\partial\{[h_i(\mathbf{r}) \otimes M(\mathbf{r})] \cdot [h_i^{\dagger}(\mathbf{r}) \otimes M(\mathbf{r})]\}}{\partial M(\boldsymbol{\rho})}$$
$$= \frac{\partial[h_i(\mathbf{r}) \otimes M(\mathbf{r})]}{\partial M(\boldsymbol{\rho})} \cdot [h_i^{\dagger}(\mathbf{r}) \otimes M(\mathbf{r})] + \frac{\partial[h_i^{\dagger}(\mathbf{r}) \otimes M(\mathbf{r})]}{\partial M(\boldsymbol{\rho})} \cdot [h_i(\mathbf{r}) \otimes M(\mathbf{r})]$$
$$= h_i(\mathbf{r} - \boldsymbol{\rho}) \cdot [h_i^{\dagger}(\mathbf{r}) \otimes M(\mathbf{r})] + h_i^{\dagger}(\mathbf{r} - \boldsymbol{\rho}) \cdot [h_i(\mathbf{r}) \otimes M(\mathbf{r})]. \tag{B4}$$

Noticing the definition of $Z(\mathbf{r})$ in Eq. (10), finally we can derive the gradient of the cost function $F\{M\}$ with respect to $M(\boldsymbol{\rho})$ as

$$\nabla F\{M\} = \frac{\partial F\{M(\mathbf{r})\}}{\partial M(\boldsymbol{\rho})} = \frac{\partial \|Z - Z^*\|_2^2}{\partial M(\boldsymbol{\rho})} = \sum_{\mathbf{r}} (Z - Z^*) \cdot \frac{\partial Z}{\partial M(\boldsymbol{\rho})} = a \cdot \sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot \frac{\partial \sum_i \mu_i I_i}{\partial M(\boldsymbol{\rho})}$$

$$= a \cdot \sum_i \mu_i \cdot \left[\sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot \frac{\partial I_i}{\partial M(\boldsymbol{\rho})} \right]$$

$$= a \cdot \sum_i \mu_i \cdot \left\{ \begin{array}{l} \sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i^{\dagger} \otimes M) \cdot h_i(\mathbf{r} - \boldsymbol{\rho}) \\ + \sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i \otimes M) \cdot [h_i(\mathbf{r} - \boldsymbol{\rho})]^{\dagger} \end{array} \right\}$$

$$= a \cdot \sum_i \mu_i \cdot \left\{ \begin{array}{l} \sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i^{\dagger} \otimes M) \cdot h_i^{flip}(\boldsymbol{\rho} - \mathbf{r}) \\ + \sum_{\mathbf{r}} (Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i \otimes M) \cdot [h_i^{flip}(\boldsymbol{\rho} - \mathbf{r})]^{\dagger} \end{array} \right\}$$

$$= a \cdot \sum_i \mu_i \cdot \left\{ \begin{array}{l} h_i^{flip} \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i^{\dagger} \otimes M)] \\ + (h_i^{flip})^{\dagger} \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i \otimes M)] \end{array} \right\}$$

$$= a \left\{ \sum_{i=1}^{Q} \mu_i h_i^{flip} \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i^{\dagger} \otimes M)] \right\} + a \left\{ \sum_{i=1}^{Q} (\mu_i h_i^{flip})^{\dagger} \otimes [(Z - Z^*) \cdot Z \cdot (\mathbf{1} - Z) \cdot (h_i \otimes M)] \right\}.$$

(B5)

APPENDIX C: DERIVATION OF EQ. (33)

Based on Eq. (32), the gradient of the TV regularization $R(M)$ is

$$\nabla R(M) = \nabla \int |\nabla M| dr = \nabla \int \sqrt{M_x^2 + M_y^2} \, dx dy$$

$$= -\frac{\partial}{\partial x} \frac{M_x}{\sqrt{M_x^2 + M_y^2}} - \frac{\partial}{\partial y} \frac{M_y}{\sqrt{M_x^2 + M_y^2}}$$

$$= -\nabla \cdot \left(\frac{\nabla M}{|\nabla M|} \right). \quad (C1)$$

Here, $M_x = \partial M / \partial x$ and $M_y = \partial M / \partial y$. In order reduce the value of $R(M)$ in the iteration process, the evolution velocity should be along the negative direction of $\nabla R(M)$, that leads to Eq. (33)

$$\sigma(\mathbf{r}, t) = \nabla \cdot \left(\frac{\nabla M}{|\nabla M|} \right). \quad (C2)$$

[1]A. B. Kahng and Y. C. Pati, Proc. ISPD 112 (1999).
[2]A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography* (SPIE, Bellingham, Washington, 2001).
[3]L. W. Liebmann, S. M. Mansfield, A. K. Wong, M. A. Lavin, W. C. Leipold, and T. G. Dunham, IBM J. Res. Dev. **45**, 651 (2001).
[4]F. Schellenberg, Proc. SPIE **5377**, 1 (2004).
[5]Y. Liu and A. Zakhor, Proc. SPIE **1264**, 401 (1990).
[6]Y. Liu and A. Zakhor, Proc. SPIE **1463**, 382 (1991).
[7]Y. Liu and A. Zakhor, IEEE Trans. Semicond. Mater. **5**, 138 (1992).
[8]Y. C. Pati and T. Kailath, J. Opt. Soc. Am. A **11**, 2438 (1994).
[9]S. Sherif, B. Saleh, and R. De Leone, IEEE Trans. Image Process. **4**, 1252 (1995).
[10]Y. Granik, Proc. SPIE **5754**, 506 (2005).
[11]A. Poonawala and P. Milanfar, Proc. SPIE **5674**, 114 (2005).
[12]A. Poonawala and P. Milanfar, IEEE Trans. Image Process **16**, 774 (2007).
[13]A. Poonawala and P. Milanfar, Microelectron. Eng. **84**, 2837 (2007).
[14]X. Ma and G. R. Arce, Opt. Express **15**, 15066 (2007).
[15]X. Ma and G. R. Arce, J. Opt. Soc. Am. A **25**, 2960 (2008).
[16]X. Ma and G. R. Arce, Opt. Express **16**, 20126 (2008).
[17]X. Ma and G. R. Arce, Opt. Express **17**, 5783 (2009).
[18]X. Ma and G. R. Arce, Opt. Express **19**, 2165 (2011).
[19]X. Ma and G. R. Arce, J. Opt. Soc. Am. A **26**, 1687 (2009).
[20]X. Ma, G. R. Arce, and Y. Li, Appl. Opt. **50**, 5567 (2011).
[21]X. Ma, Y. Li, and L. Dong, J. Opt. Soc. Am. A **29**, 1300 (2012).
[22]S. H. Chan, A. K. Wong, and E. Y. Lam, Opt. Express **16**, 14746 (2008).
[23]J. Li, Y. Shen, and E. Y. Lam, Opt. Express **20**, 21792 (2012).
[24]J. Li, S. Y. Liu, and E. Y. Lam, Opt. Express **21**, 8076 (2013).
[25]N. Jia, A. K. Wong, and E. Y. Lam, Proc. SPIE **7140**, 71401W (2008).
[26]N. Jia and E. Y. Lam, J. Opt. **12**, 045601 (2010).
[27]N. Jia and E. Y. Lam, Opt. Express **19**, 19384 (2011).
[28]S. K. Choy, N. Jia, C. S. Tong, M. L. Tang, and E. Y. Lam, SIAM J. Imaging Sci. **5**, 625, (2012).
[29]E. Y. Lam and A. K. Wong, Opt. Express **17**, 12259 (2009).
[30]J. C. Yu and P. Yu, Opt. Express **18**, 23331 (2010).
[31]S. Osher and R. P. Fedkiw, J. Comput. Phys. **169**, 463 (2001).
[32]S. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces* (Springer, New York, 2002).
[33]L. Pang, Y. Liu, and D. Abrams, Proc. SPIE **6607**, 660739 (2007).
[34]L. Pang, G. Dai, T. Cecil, T. Dam, Y. Cui, P. Hu, D. Chen, K. Baik, and D. Peng, Proc. SPIE **6924**, 69240T (2008).
[35]Y. Shen, N. Wong, and E. Y. Lam, Opt. Express **17**, 23690 (2009).
[36]Y. Shen, N. Jia, N. Wong, and E. Y. Lam, Opt. Express **19**, 5511 (2011).
[37]J. Nocedal and S. J.Wright, *Numerical Optimization*, 2nd ed. (Springer, New York, 2006), Chaps. 3 and 5.
[38]P. Gong, S. Y. Liu, W. Lv, and X. J. Zhou, J. Vac. Sci. Technol. B **30**, 06FG03 (2012).
[39]A. K. Wong, *Optical Imaging in Projection Microlithography* (SPIE, Bellingham, Washington, 2005).
[40]H. H. Hopkins, Proc. R. Soc. A **217**, 408 (1953).
[41]E. Polak and G. Ribière, Rev. Francaise Informat Recherche Opertionelle **3**, 35 (1969), available at http://www.numdam.org/item?id=M2AN_1969__3_1_35_0.
[42]W. W. Hager and H. Zhang, Pacific J. Optim. **2**, 35 (2006), available at http://www.ybook.co.jp/online2/oppjo/vol2/p35.html.
[43]J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations* (SIAM, Philadelphia, 2004).
[44]A. Harten, B. Engquist, S. Osher, and S. Chakravarthy, J. Comput. Phys. **71**, 231 (1987).
[45]F. Santosa, ESAIM Contröle Optim. Calc. Var. **1**, 17 (1996).
[46]A. Marquina and S. Osher, SIAM J. Sci. Comput. **22**, 387 (2000).
[47]W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University, Cambridge, England, 2007), Chap. 10.2.
[48]N. Cobb, "Fast optical and process proximity correction algorithms for integrated circuit manufacturing," Ph.D. dissertation (University of California at Berkeley, 1998).
[49]S. Y. Liu, X. F. Wu, W. Liu and C. W. Zang, J. Vac. Sci. Technol. B **29**, 06FH03 (2011).
[50]W. T. Pong and A. K. Wong, Proc. SPIE **4691**, 169 (2002).