Optics EXPRESS

Nonlinear compressive inverse lithography aided by low-rank regularization

XU MA,[1,*] ZHIQIANG WANG,[1] JIANCHEN ZHU,[2] SHENGEN ZHANG,[1] GONZALO R. ARCE,[3] AND SHENGJIE ZHAO[4]

[1] *Key Laboratory of Photoelectronic Imaging Technology and System of Ministry of Education of China, School of Optics and Photonics, Beijing Institute of Technology, Beijing, 100081, China*
[2] *College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China*
[3] *Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA*
[4] *School of Software Engineering, Tongji University, Shanghai, 201804, China*
[*] *maxu@bit.edu.cn*

Abstract: Photolithography is at the core of the semiconductor industry that is used to fabricate microscale and nanoscale integrated circuits. Inverse lithography is a technique extensively used to compensate for lithography patterning distortions. It refers to methods that pre-distort the photomask patterns such that their projection, through the photolithography system, results in a pattern that is as close as possible to the intended original. However, most inverse lithography technique (ILT) methods suffer from large computational complexity. This paper develops a nonlinear compressive sensing framework for ILT that effectively improves the computational efficiency and image fidelity, while at the same time controlling the mask complexity. Based on a nonlinear lithography imaging model, the compressive ILT is formulated as an inverse optimization problem aimed at reducing the patterning error, and enforcing the sparsity and low rank properties of the mask pattern. A downsampling strategy is adopted to reduce the dimensionality of the cost function, thus alleviating the computational burden. Sparsity and low-rank regularizations are then used to constrain the solution space and reduce the mask complexity. The split Bregman algorithm is used to solve for the inverse optimization problem. The superiority of the proposed method is verified by a set of simulations and comparison to traditional ILT algorithms.

1. Introduction

Large scale integrated circuits to date are made primarily by photolithography, a process similar to photography, where the photomask is used to print the layouts of integrated circuits onto the silicon wafers [1,2]. The sketch of a typical deep ultraviolet photolithography system is shown in Fig. 1. The 193nm argon fluoride light source is used to illuminate the mask that contains the layout patterns. Then, the image of the mask is transferred through the projector onto the substrate coated by the photoresist. Photoresist consists of light-sensitive materials, such that the chemical properties of the exposed areas are changed. Thus, after a series of photoresist development processes, the layout pattern is printed on the wafer. From a mathematical point of view, the lithography imaging process can be described as a nonlinear function, where the mask pattern and print image are the input and output, respectively. Ideally, the output print image on the wafer should be identical to the mask pattern. However, the lithography process introduces imaging distortions due to the optical limits, such as the effects of diffraction, interference and so on [3,4].

In order to improve the resolution and image fidelity of lithography systems, inverse lithography techniques (ILT) are used to counteract the image distortions. ILT methods aim at inverting the imaging model of lithography so as to optimize the mask pattern in such a way that the distortion is pre-compensated before the image is printed [5,6]. Different from the conventional rule-based

#373672
Journal © 2019
https://doi.org/10.1364/OE.27.029992
Received 26 Jul 2019; accepted 12 Sep 2019; published 2 Oct 2019

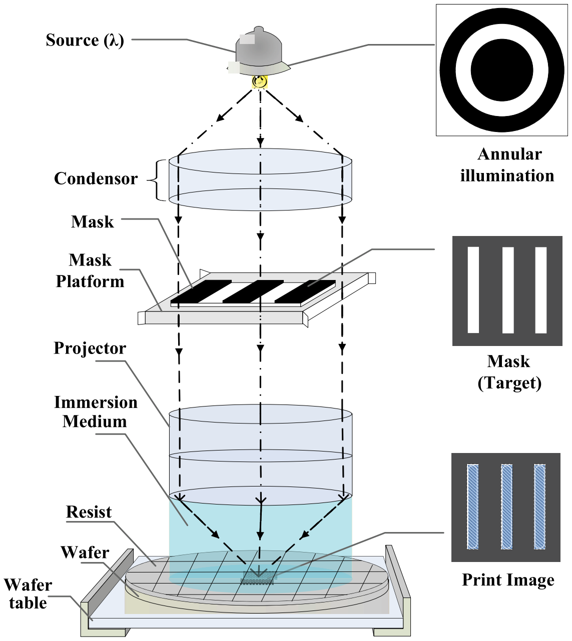


Fig. 1. Sketch of the deep ultraviolet photolithography system.

or edge-based mask optimization methods, ILTs pixelate the mask pattern and optimize all mask pixels [7–9]. ILT approaches can effectively improve the imaging performance because of the higher degrees of optimization freedom [7–19]. However, current ILT methods are mostly gradient-based, which need to calculate the derivatives of cost functions with respect to all mask pixels in each iteration, thus resulting in excessive computational complexity, especially for sophisticated large-scale mask patterns. In addition, ILT methods inevitably increase the complexity of the mask patterns attributed to the inserted sub-resolution assist features and curvilinear boundaries of mask features [10–12]. Trapezoid count of the fractured mask pattern is a widely used metric to assess the complexity and fabrication cost of the mask pattern [13]. Each trapezoid is associated to at least one shot in the mask writing machine, thus the trapezoid count will affect the mask writing time and fabrication cost [14]. Therefore, the mask complexity is another important issue to be considered in ILT optimization procedures.

In last several years, the principles of compressive sensing (CS) have been applied to improve the speed of light source optimization (SO) methods in photolithography [20,21]. Given a fixed mask pattern, SO approaches can effectively improve the process window of lithography systems by iteratively optimizing the illumination arrays. Based on the linear mapping between the illumination array and lithographic aerial image, conventional linear CS methods can be used to significantly reduce the dimensionality of the SO problem [22,23]. However, ILT approaches aim at optimizing the mask pattern instead of the illumination array, which is described using a nonlinear mathematical model. Thus, nonlinear CS methods have to be applied to solve for the ILT optimization problem [24–26]. Recent advances in nonlinear CS theory makes it possible to reconstruct a sparse signal from a small set of nonlinear-mapping measurements of the underlying signal. Recently, Ma et al. applied the iterative hard threshold (IHT) method proposed by Blumensath and Davies [27–29] to accelerate the speed of ILT algorithms based on the concept of nonlinear CS [30]. However, the lithography imaging performance obtained by this method remains to be improved, since the hard thresholding operation likely leads to suboptimal solutions. What is more, the mask complexity is not explicitly involved in the cost function of IHT algorithm.

This work establishes a more general nonlinear CS framework for inverse lithography, which improves the computational efficiency and image fidelity over the traditional gradient-based ILT method, as well as controlling the trapezoid count of fractured mask pattern. As previously mentioned, the lithography imaging model is nonlinear, and the cost function is established based on the patterning error defined as the difference between the target layout and the actual print image on the wafer. The layout pattern is then downsampled on a sparse mesh to reduce the dimensionality of the cost function, and alleviate the computing burden of ILT optimization algorithm. Although the number of optimization variables still include all mask pixels, the computational complexity to calculate the gradients of cost function on the sparse mesh is lower than that on the dense mesh. The downsampling makes the ILT problem underdetermined, and nonlinear CS methods are used to solve the inverse problem when rank, intensity and sparsity priors are used in the inverse optimization problem to reduce the patterning error, and enforce the sparsity and low-rank properties of the mask pattern [31–33]. It is noted that downsampling directly reduces the computational complexity of the ILT optimization algorithm. The sparsity and low-rank regularizations can help to recover the optimized mask patterns from the underdetermined problem. In addition, these regularizations are beneficial for controlling the mask complexity. The split Bregman algorithm is used to efficiently solve for the inverse optimization problem [34], and the computational complexity is analyzed. With the help of auxiliary variables, the split Bregman algorithm can also improve the convergence characteristics compared to the IHT algorithm [30]. Finally, the proposed method is assessed by a set of simulations, and compared to traditional ILT algorithms. The proposed method effectively improves the computational speed, and achieves superior imaging performance over other methods.

The remainder of this paper is organized as follows. First, the nonlinear imaging model of photolithography system is described in Section 2. Then, the compressive ILT optimization problem is formulated in Section 3. The ILT optimization algorithm based on split Bregman method is described in Section 4. Simulations and analysis are presented in Section 5. Finally, the conclusions are provided in Section 6.

2. Nonlinear imaging model of photolithography system

The imaging process of photolithography systems can be characterized with a nonlinear model, where the input is the mask pattern $\mathbf{M}$ and the output is the print image $\mathbf{Z}$ on the wafer. Let $\mathbf{M}$ be represented by an $N \times N$ real matrix, denoted by $\mathbf{M} \in \mathbb{R}^{N\times N}$. $\mathbf{J} \in \mathbb{R}^{N_s\times N_s}$ represents the intensity distribution of the illumination array, $\mathbf{H}_p \in \mathbb{R}^{N\times N}$ is the blur matrix representing the point spread function of the lithography system along the p-axis ($p = x,\ y$, or z), and $\mathbf{B} \in \mathbb{R}^{N\times N}$ is the matrix representing the oblique incidence effect of the light rays on the mask plane. The aerial image $\mathbf{I}$ projected on the wafer can be calculated by Abbe's method as [35,36]

$$\mathbf{I} = \frac{1}{J_{\text{sum}}} \sum_{x_s} \sum_{y_s} \left[\mathbf{J}(x_s, y_s) \times \sum_{p=x,y,z} |\mathbf{H}_p^{x_s y_s} \otimes (\mathbf{B}^{x_s y_s} \odot \mathbf{M})|^2 \right], \tag{1}$$

where $\mathbf{J}(x_s, y_s)$ is the intensity of the source point at coordinate (x_s, y_s) on the source plane, and $J_{\text{sum}} = \sum_{x_s} \sum_{y_s} \mathbf{J}(x_s, y_s)$ is the normalization factor. The upperscript "$x_s y_s$" means the matrices $\mathbf{H}_p^{x_s y_s}$ and $\mathbf{B}^{x_s y_s}$ are the functions of the coordinate (x_s, y_s). The notation $\otimes$ and $\odot$ represent the convolution operation and the entry-by-entry multiplication, respectively.

After the photoresist is developed, the aerial image is transferred to the print image $\mathbf{Z}$. The photoresist effect can be represented by a hard threshold function [6,8], and the print image is formulated as

$$\mathbf{Z} = \Gamma\{\mathbf{I} - t_r\}, \tag{2}$$

where $\mathbf{I}$ is described in Eq. (1), and t_r is the threshold value of the photoresist. $\Gamma\{\cdot\} = 1$ if the argument is larger than 0, otherwise $\Gamma\{\cdot\} = 0$. The iterative optimization algorithm requires the gradient of the cost function, thus we use the differentiable sigmoid function to approximate the hard threshold function. The sigmoid function is

$$\mathrm{sig}(x, t_r) = \frac{1}{1 + \exp[-a(x - t_r)]}, \tag{3}$$

where t_r is the process threshold, and a dictates the steepness of the sigmoid function. Using the sigmoid function, Eq. (2) can be approximated to $\mathbf{Z} = \mathrm{sig}\{\mathbf{I}, t_r\}$.

3. Formulation of compressive ILT optimization problem

3.1. Optimization framework of compressive ILT

In this section, the optimization framework of compressive ILT is derived. The ILT problem seeks to find the optimal mask pattern $\hat{\mathbf{M}}$ that minimizes the cost function $f(\mathbf{M})$, which is defined as the l_2-norm of the difference between the print image $\mathbf{Z}$ and the target layout $\tilde{\mathbf{Z}}$. The target layout is the ideal image expected to be printed on the wafer. Thus, the ILT problem can be formulated as

$$\begin{aligned}\hat{\mathbf{M}} &= \arg\min_{\mathbf{M}} f(\mathbf{M}) = \arg\min_{\mathbf{M}} \|\mathbf{\Pi} \odot (\tilde{\mathbf{Z}} - \mathbf{Z})\|_2^2 \\ &= \arg\min_{\mathbf{M}} \sum_{m=1}^{N}\sum_{n=1}^{N} \left\{\mathbf{\Pi}(m,n) \times \left[\tilde{\mathbf{Z}}(m,n) - \mathbf{Z}(m,n)\right]\right\}^2,\end{aligned} \tag{4}$$

where $\mathbf{\Pi} \in \mathbb{R}^{N\times N}$ is a weight matrix used to emphasize or deemphasize the patterning errors in different layout regions [12]. $\mathbf{\Pi}(m,n)$, $\tilde{\mathbf{Z}}(m,n)$ and $\mathbf{Z}(m,n)$ represent the (m,n)th elements in $\mathbf{\Pi}$, $\tilde{\mathbf{Z}}$ and $\mathbf{Z}$, respectively. It is observed from Eqs. (1) and (2) that the aerial image is a quadratic function of the mask pattern, while the print image is a hard threshold function of the aerial image. Furthermore, the cost function in Eq. (4) is a quadratic function of the print image. Thus, the cost function is nonlinear to the mask pattern.

Traditional gradient-based ILT algorithms iteratively calculate the gradient of the cost function and update the mask patterns. Based on the pixelated imaging model, the gradient has to be calculated many times on the dense mesh. Computing the gradient of the cost function is a time consuming procedure, and the computational complexity of ILT algorithms increases exponentially with the size of the mask. To overcome this limitation, this paper adopts the layout downsampling method to reduce computational complexity of the gradient [30].

Suppose we grid the layout pattern on a sparse mesh. Let $\tilde{\mathbf{Z}}_K$, $\mathbf{Z}_K$ and $\mathbf{\Pi}_K \in \mathbb{R}^{N/K\times N/K}$ be the downsampled versions of the target layout, print image and weight matrix, respectively. K is the downsampling rate. In particular, $\tilde{\mathbf{Z}}_K(m,n) = \tilde{\mathbf{Z}}(Km,Kn)$, $\mathbf{Z}_K(m,n) = \mathbf{Z}(Km,Kn)$ and $\mathbf{\Pi}_K(m,n) = \mathbf{\Pi}(Km,Kn)$. The cost function on the sparse mesh is redefined as

$$\begin{aligned}f_K(\mathbf{M}) &= \|\mathbf{\Pi}_K \odot (\tilde{\mathbf{Z}}_K - \mathbf{Z}_K)\|_2^2 \\ &= \sum_{m=1}^{N/K}\sum_{n=1}^{N/K} \left\{\mathbf{\Pi}(Km,Kn) \times \left[\tilde{\mathbf{Z}}(Km,Kn) - \mathbf{Z}(Km,Kn)\right]\right\}^2.\end{aligned} \tag{5}$$

The downsampling operation reduces the complexity to calculate the cost function and its gradient.

In practice, the lithographic imaging performance is always influenced by process variations. This paper considers two kinds of process variations referred to as the defocus and dose variation.

Defocus happens when the actual wafer plane deviates from the focal plane due to the wafer topography, system vibration and so on. On the other hand, dose variation represents the deviation of the actual exposure dose of illuminations from the nominal dose. The effects of defocus and dose variation may degrade the image fidelity that is projected using the mask pattern optimized under nominal conditions. In order to obtain image fidelity robust to process variations, a loss term is added into the cost function to take into account the effects of defocus and dose variation. Thus, the cost function is finally modified as

$$d(\mathbf{M}) = \alpha f_K(\mathbf{M}) + (1-\alpha) f'_K(\mathbf{M}), \tag{6}$$

where α is the weight parameter used to balance the focal and defocus imaging performance, $f_K(\mathbf{M})$ is the same as Eq. (5), and $f'_K(\mathbf{M})$ represents the loss term on the defocus plane, which is defined as

$$f'_K(\mathbf{M}) = \|\mathbf{\Pi}_K \odot (\tilde{\mathbf{Z}}_K - \mathbf{Z}'_K)\|_2^2, \tag{7}$$

where $\mathbf{Z}'_K \in \mathbb{R}^{N/K \times N/K}$ is the downsampled print image on the defocus plane diverged from the focal plane by δnm. According to Eqs. (2) and (3), $\mathbf{Z}' = \text{sig}\{\gamma \mathbf{I}, t_r\}$ where γ is an amplitude modulation coefficient to emulate the dose variation of the light source. In the following simulations, $\alpha = 0.5$, $\delta = 10$nm, and $\gamma = 1.1$, which means the exposure is 10% over the nominal dose. In real applications, the parameters α, δ and γ can be adjusted according to the specific simulation cases. The first loss term in Eq. (6) aims at reducing the patterning error on the focal plane under nominal dose. The second loss term in Eq. (6) is used to reduce the patterning error on the defocus plane under dose variation. Minimizing the cost function in Eq. (6) means improving the robustness of the photolithography system in a range of defocus and dose variation, thus effectively extending the process window.

During the optimization process, the mask pixels are constrained to several discrete values. For instance, each pixel value of the binary mask is chosen as 0 or 1 [4]. Thus, Eq. (4) is a discrete and bound-constrained optimization problem. In order to use the gradient-based algorithm to solve it, we relax Eq. (4) to an unconstrained optimization problem by using the following parametric transformation:

$$\mathbf{M} = 0.5 \times (1 + \cos \mathbf{\Theta}), \tag{8}$$

where $\mathbf{M}$ is the mask pattern, and $\mathbf{\Theta} \in \mathbb{R}^{N \times N}$ is a parameter matrix with entry values in the range of $(-\infty, \infty)$. Regardless of $\mathbf{\Theta}$, the mask pixel is alway in the interval of [0,1]. Thus, the optimization variables are transferred from $\mathbf{M}$ to $\mathbf{\Theta}$, and the ILT problem is reformulated as

$$\hat{\mathbf{\Theta}} = \arg \min_{\mathbf{\Theta}} d(\mathbf{\Theta}), \tag{9}$$

where the cost function d is described in Eq. (6).

3.2. Regularizations

After the downsampling operation, the ILT problem becomes underdetermined. Our numerical experiments suggest that the proposed nonlinear CS can help reconstruct the optimized mask patterns from a few measurements under the assumption that the masks can be sparsely represented on a set of predefined basis functions [24–26]. Following the sparse representation, the coefficient matrix $\mathbf{\Theta}$ in Eq. (8) can be written as

$$\mathbf{\Theta} = \mathbf{\Psi} \bar{\mathbf{\Theta}} \mathbf{\Psi}^T, \tag{10}$$

where $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ is the transform matrix of the two-dimensional (2D) representation basis, and $\bar{\mathbf{\Theta}} \in \mathbb{R}^{N \times N}$ denotes the sparse coefficients on the basis. For the orthonormal basis, the sparse coefficients can be calculated as $\bar{\mathbf{\Theta}} = \mathbf{\Psi}^T \mathbf{\Theta} \mathbf{\Psi}$. In the following, the 2D discrete cosine transform

(DCT) basis is selected as the representation basis, since the DCT proves to have good energy compaction characteristics across a wide-variety of image types. Other representation bases, such as the Haar wavelet basis and the discrete Fourier transform (DFT) basis can be straightforwardly used.

Inspired by the PRISM model proposed in [31] and [32], the sparsity and low-rank properties of the mask are exploited to aid in solving the underdetermined ILT optimization problem. In particular, we add a sparse penalty term and a low-rank penalty term in the cost function to reduce the solution space, and the cost function in Eq. (6) is adjusted as

$$d(\mathbf{\Theta}) = \alpha f_K(\mathbf{\Theta}) + (1-\alpha) f'_K(\mathbf{\Theta}) + \lambda_* \|\mathbf{\Theta}\|_* + \lambda_1 \|\bar{\mathbf{\Theta}}\|_1, \tag{11}$$

where λ_1 and λ_* are the weights for the sparsity and low-rank regularizations. $\|\cdot\|_1$ is the l_1-norm that constrains the sparsity degree of $\mathbf{\Theta}$, and $\|\cdot\|_*$ is the nuclear norm for the rank regularization on $\mathbf{\Theta}$, which is defined as

$$\|\mathbf{\Theta}\|_* = \sum_{i=1}^{r} \sigma_i, \tag{12}$$

where σ_i is the ith largest singular value of $\mathbf{\Theta}$, r is the rank of $\mathbf{\Theta}$. It is known that images with lower rank include more replicated or similar local regions. Thus, reducing the rank of the matrix $\mathbf{\Theta}$ will help to remove the randomly distributed pixels that are not similar or correlated to other mask features. In addition, the sparsity regularization will constrain the number of significant frequency components in the mask pattern. Thus, the use of both sparsity and low rank penalty terms is expected to reduce the complexity of the optimized mask patterns.

We also employ the quadratic penalty term $R_q(\mathbf{\Theta})$ in [8] to ensure that the optimized mask pattern is restricted to be binary. Additionally, we adopt the wavelet penalty term $R_w(\mathbf{\Theta})$ in [6] to reduce the complexity of the mask pattern. Thus, the overall cost function including all penalty terms is given by

$$d(\mathbf{\Theta}) = \alpha f_K(\mathbf{\Theta}) + (1-\alpha) f'_K(\mathbf{\Theta}) + \lambda_* \|\mathbf{\Theta}\|_* + \lambda_1 \|\bar{\mathbf{\Theta}}\|_1 + \lambda_q R_q(\mathbf{\Theta}) + \lambda_w R_w(\mathbf{\Theta}), \tag{13}$$

where λ_q and λ_w are the weights of the quadratic penalty and wavelet penalty, respectively. The formula of $R_q(\mathbf{\Theta})$ and $R_w(\mathbf{\Theta})$ can be found in Appendix A. In the end, the compressive ILT optimization problem can be formulated as

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \left\{ \alpha f_K(\mathbf{\Theta}) + (1-\alpha) f'_K(\mathbf{\Theta}) + \lambda_* \|\mathbf{\Theta}\|_* + \lambda_1 \|\bar{\mathbf{\Theta}}\|_1 + \lambda_q R_q(\mathbf{\Theta}) + \lambda_w R_w(\mathbf{\Theta}) \right\}. \tag{14}$$

4. Solution of ILT problem based on split Bregman algorithm

The split Bregman algorithm is usually used to solve the linear reconstruction problems, where the optimization problem is separated into a series of sub-problems that can be solved efficiently. This paper was inspired by the principles behind the traditional split Bregman algorithm, and a modified split Bregman algorithm is developed to solve the compressive ILT optimization problem in Eq. (14). In order to improve the optimization performance, we simplify the l_2-norm to the add-residual-back iterative scheme by introducing auxiliary variables $\mathbf{E}_K$ and $\mathbf{E}'_K$ [34,37]. To efficiently handle the non-differentiable norms, we replace the arguments of l_1-norm and nuclear norm by introducing additional auxiliary variables $\mathbf{D} \approx \mathbf{\Theta}$ and $\mathbf{C} \approx \bar{\mathbf{\Theta}}$ [31,32]. Then, we use the similar add-residual-back iterative scheme to relax l_1-norm into l_2-norm by introducing the auxiliary variable $\mathbf{V}$ representing the difference between $\mathbf{D}$ and $\mathbf{\Theta}$, and the auxiliary variable $\mathbf{W}$ representing the difference between $\mathbf{C}$ and $\bar{\mathbf{\Theta}}$. Finally, the ILT optimization problem is reduced to

$$\begin{aligned}\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \big\{&\alpha\|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}_K(\boldsymbol{\Theta}) + \mathbf{E}_K]\|_2^2 + (1-\alpha)\|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}'_K(\boldsymbol{\Theta}) \\ &+\mathbf{E}'_K]\|_2^2 + \lambda_*\|\mathbf{D}\|_* + \mu_*\|\mathbf{D}-\boldsymbol{\Theta}-\mathbf{V}\|_2^2 + \lambda_1\|\mathbf{C}\|_1 + \mu_1\|\mathbf{C}-\bar{\boldsymbol{\Theta}}-\mathbf{W}\|_2^2 \\ &+\lambda_q R_q(\boldsymbol{\Theta}) + \lambda_w R_w(\boldsymbol{\Theta})\big\},\end{aligned} \tag{15}$$

where μ_* and μ_1 are the regularization parameters.

The solution of Eq. (15) can be separated into the following four steps. In the optimization process, the coefficients $\boldsymbol{\Theta}$ and $\bar{\boldsymbol{\Theta}}$ are updated iteratively.

Step 1. Update the coefficient matrix $\boldsymbol{\Theta}$ and the auxiliary variables $\mathbf{E}_K$ and $\mathbf{E}'_K$:

$$\begin{aligned}\boldsymbol{\Theta}^{n+1} = \arg\min_{\boldsymbol{\Theta}} \big\{&\alpha\|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}_K(\boldsymbol{\Theta}^n) + \mathbf{E}_K^n]\|_2^2 + (1-\alpha)\|\boldsymbol{\Pi}_K \\ &\odot[\tilde{\mathbf{Z}}_K - \mathbf{Z}'_K(\boldsymbol{\Theta}^n) + \mathbf{E}_K^{\prime n}]\|_2^2 + \mu_*\|\mathbf{D}^n - \boldsymbol{\Theta}^n - \mathbf{V}^n\|_2^2 \\ &+\mu_1\|\mathbf{C}^n - \bar{\boldsymbol{\Theta}}^n - \mathbf{W}^n\|_2^2 + \lambda_q R_q(\boldsymbol{\Theta}^n) + \lambda_w R_w(\boldsymbol{\Theta}^n)\big\},\end{aligned} \tag{16}$$

$$\mathbf{E}_K^{n+1} = \mathbf{E}_K^n + \tilde{\mathbf{Z}}_K - \mathbf{Z}_K(\boldsymbol{\Theta}^{n+1}), \tag{17}$$

$$\mathbf{E}_K^{\prime n+1} = \mathbf{E}_K^{\prime n} + \tilde{\mathbf{Z}}_K - \mathbf{Z}'_K(\boldsymbol{\Theta}^{n+1}), \tag{18}$$

where n is the iteration number.

Step 2. Update the auxiliary variables $\mathbf{D}$ and $\mathbf{V}$:

$$\mathbf{D}^{n+1} = \arg\min_{\mathbf{D}} \lambda_*\|\mathbf{D}^n\|_* + \mu_*\|\mathbf{D}^n - (\boldsymbol{\Theta}^{n+1} + \mathbf{V}^n)\|_2^2, \tag{19}$$

$$\mathbf{V}^{n+1} = \mathbf{V}^n + \boldsymbol{\Theta}^{n+1} - \mathbf{D}^{n+1}. \tag{20}$$

Step 3. Update the auxiliary variables $\mathbf{C}$ and $\mathbf{W}$:

$$\mathbf{C}^{n+1} = \arg\min_{\mathbf{C}} \lambda_1\|\mathbf{C}^n\|_1 + \mu_1\|\mathbf{C}^n - (\bar{\boldsymbol{\Theta}}^{n+1} + \mathbf{W}^n)\|_2^2, \tag{21}$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \bar{\boldsymbol{\Theta}}^{n+1} - \mathbf{C}^{n+1}. \tag{22}$$

Step 4. If the algorithm converges, terminate the algorithm, otherwise, return **Step 1**.

The cost function in Eq. (16) is a quadratic cost function that can be minimized by the steepest descent (SD) algorithm. Thereafter, we need to calculate the gradient of the cost function with respect to $\boldsymbol{\Theta}$, which is described in Appendix B. Since $\boldsymbol{\Theta}$ is correlated to other optimization variables in Eqs. (17)–(22), updating all optimization variables together is beneficial to avoid the local minima of the cost function. Thus, we just update $\boldsymbol{\Theta}$ once in Eq. (16) based on the gradient of the cost function, and then successively update other variables in Eqs. (17)–(22). After that, the algorithm will enter the next loop to update $\boldsymbol{\Theta}$ again.

According to [32] and [38], the solution in Eq. (19) is given by

$$\mathbf{D}^{n+1} = S_\tau(\boldsymbol{\Theta}^{n+1} + \mathbf{V}^n) \tag{23}$$

where $\tau = \lambda_*/\mu_*$, and $S_\tau\{\cdot\}$ is the soft-thresholding singular value decomposition operator. For an arbitrary square matrix $\mathbf{X} \in \mathbb{R}^{N\times N}$, the operator $S_\tau\{\cdot\}$ is defined as following. The singular value decomposition of $\mathbf{X}$ with rank r is

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{Q}^T, \tag{24}$$

where $\Sigma = \mathrm{diag}(\{\sigma_i\}_{1\le i\le r})$ is an $r \times r$ diagonal matrix and σ_i is the ith singular value of $\mathbf{X}$. Both $\mathbf{U}$ and $\mathbf{Q}$ are the $N \times r$ matrices with orthonormal columns. Define another $r \times r$ diagonal matrix $\Sigma' = \text{diag}(\max\{\sigma_i - \tau, 0\})$, which is the thresholding version of Σ. Note that the diagonal element of Σ' is set as $\sigma_i - \tau$ if $\sigma_i > \tau$. Otherwise, the diagonal element of Σ' is set to be zero. Then, the soft-thresholding operator on $\mathbf{X}$ is defined as

$$S_\tau(\mathbf{X}) = \mathbf{U}\Sigma'\mathbf{Q}^T. \tag{25}$$

According to [34], the problem in Eq. (21) can be solved as

$$\mathbf{C}^{n+1} = \Lambda\{\bar{\mathbf{\Theta}}^{n+1} + \mathbf{W}^n, \lambda_1/\mu_1\}, \quad \lambda_1/\mu_1 \geq 0, \tag{26}$$

where $\bar{\mathbf{\Theta}}^{n+1} = \mathbf{\Psi}^T\mathbf{\Theta}^{n+1}\mathbf{\Psi}$, and $\Lambda\{\cdot,\cdot\}$ is the shrink operator. For an arbitrary matrix $\mathbf{X} \in \mathbb{R}^{N\times N}$, the operator is defined as

$$\Lambda\{\mathbf{X}, \lambda_1/\mu_1\} = \frac{\mathbf{X}}{\bar{\mathbf{X}}} \odot \mathbf{X}^*, \tag{27}$$

where $\bar{\mathbf{X}} = \sqrt{\sum_{i=1}^N \sum_{j=1}^N \mathbf{X}(i,j)^2}$, $\mathbf{X}(i,j)$ represents the (i,j)th element of $\mathbf{X}$ and the (i,j)th element in $\mathbf{X}^* \in \mathbb{R}^{N\times N}$ is given by $\max\{\mathbf{X}(i,j) - \lambda_1/\mu_1, 0\}$.

Next, we make the analysis on the computational complexity of the proposed method, and compare it to traditional ILT methods based on the SD algorithm [6] and the IHT algorithm [30]. In these three methods, the fast Fourier transform (FFT) is used to compute the convolutions. The complexity to calculate the FFT on an $N \times N$ matrix is $O(N^2 \log N^2)$, and the complexity to calculate the entry-by-entry multiplication between two $N \times N$ matrices is $O(N^2)$. For the SD method, the complexity to calculate the print image and the gradient of the cost function in each iteration is $O\left[\tilde{N}_s(40N^2 + 20N^2 \log N^2)\right]$, where $\tilde{N}_s$ is the number of source points in the partially coherent illumination. For the IHT method, the computational complexity in each iteration is $O\left\{\tilde{N}_s\left[34N^2 + 20N^2\log(N^2/K^2) + 6N^2/K^2\right]\right\}$. For the proposed method, according to the formulae in Appendix B, the computational complexity in each iteration is $O\left\{\tilde{N}_s\left[36N^2 + 20N^2\log(N^2/K^2) + 6N^2/K^2\right] + rN^2\right\}$, where rN^2 is the computational complexity of singular value decomposition in Eq. (24), and r represents the rank of matrix $\mathbf{X}$. Therefore, the ratio of the computational complexity between the proposed method and SD method is given by

$$\begin{aligned} R_c &= \frac{O\left\{\tilde{N}_s\left[36N^2 + 20N^2\log(N^2/K^2) + 6N^2/K^2\right] + rN^2\right\}}{O\left[\tilde{N}_s(40N^2 + 20N^2\log N^2)\right]} \\ &= \frac{O\left(36 + 20\log N^2 - 20\log K^2 + 6/K^2 + r/\tilde{N}_s\right)}{O(40 + 20\log N^2)} \\ &\approx \frac{O(40\log N - 40\log K)}{O(40\log N)} \\ &= 1 - O\left(\frac{\log K}{\log N}\right). \end{aligned} \tag{28}$$

In the third line of Eq. (28), we ignore the constants and the terms of $6/K^2$ and $r/\tilde{N}_s$ since $20\log N^2$ should be much larger than $6/K^2$ and $r/\tilde{N}_s$. Compared to the SD method, the proposed method can reduce the computational complexity by a factor of $O(\log K/\log N)$. On the other hand, the proposed method and the IHT method have comparable complexity, and the speed of these two methods will be increased with the logarithm of the downsampling rate K. However, in the following we will show that the proposed method can achieve better lithographic imaging performance than the IHT method.

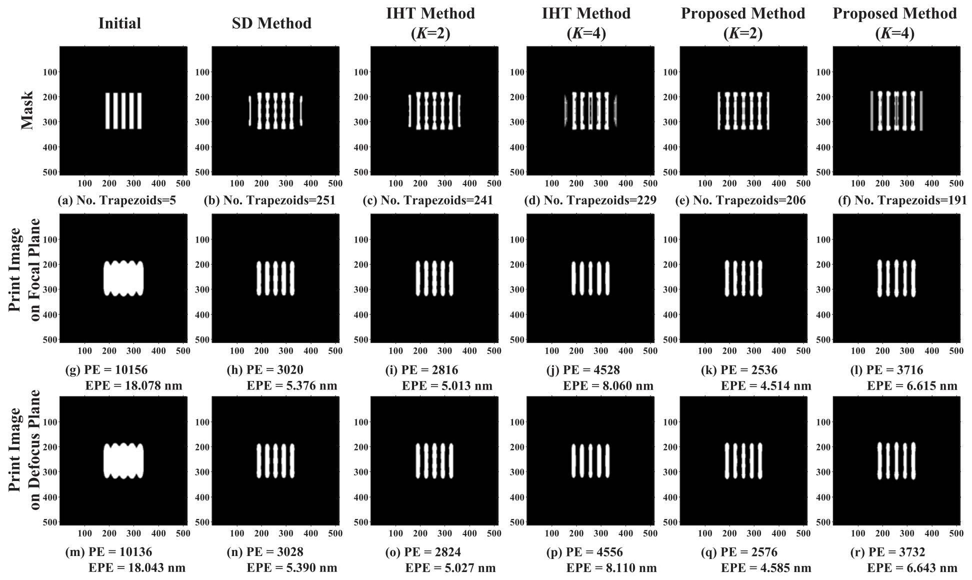
5. Simulation and analysis

This section presents a set of simulations at 45nm technology node to evaluate the performance of the proposed method, and compare it to the traditional ILT methods based on SD and IHT algorithms. The following simulations select an annular illumination with the inner and outer partial coherence factors $\sigma_{in} = 0.82$ and $\sigma_{out} = 0.97$ as the lithography light source. The wavelength of the light source is 193nm. We uniformly and symmetrically select 24 points on the source pattern to represent the annular illumination. The numerical aperture (NA) on the wafer side is 1.2, the refraction index of the immersion medium between the projector and wafer is 1.44, and the demagnification factor of the project optics is 4. In Eq. (3), we set $t_r = 0.2$ and $a = 25$. In the following, we investigate two cases with two-fold downsampling $K = 2$ and four-fold downsampling $K = 4$. To keep the edge placement accuracy of the mask features, the top boundaries and left boundaries of mask features are enforced to be sampled during the downsampling process. In addition, the mask patterns are enforced to be symmetric along the horizontal and vertical middle lines during the optimization procedure. Thus, the sparse sampling points will control the edge placement accuracy on both sides of the line features. The iteration number of the proposed method and IHT method is set to be 100, and the iteration number of the SD method is set to be 150.

The simulations based on the vertical line-space layout pattern are illustrated in Fig. 2, where black and white colors represent the pixel values of 0 and 1, respectively. The light wave emitted from the source is polarized in the Y-direction, because the Y-polarization is beneficial for improving the resolution of vertical line-space features. The target pattern is shown in Fig. 2(a), which has the critical dimension (CD) of 45nm and duty ratio of 1:1. The mask dimension is 5760nm × 5760nm with the pixel size of 11.25nm × 11.25nm. Thus, the mask pattern is represented by a 512×512 matrix. The image fidelity metrics, such as pattern error (PE), edge placement error (EPE), are presented in Fig. 2. The PE is calculated as the square of the Euclidean distance between the target layout and print image. The EPE is defined as the error of the actual printed edge position diverging from the target. In these simulations, the EPE is defined as the average value of EPEs along the entire contour of the layout features. From top to bottom, Fig. 2 shows the mask patterns, the print images on the focal plane, and the print images on the defocus plane with $\delta = 10$nm away from the focal plane. The first column illustrates the simulation results using the initial mask. The second to fourth columns show the simulation results of the SD method, and the IHT methods with downsampling rates of $K = 2$ and $K = 4$, respectively. The fifth and sixth columns illustrate the simulation results of the proposed methods with $K = 2$ and $K = 4$, respectively. During the ILT optimization process, we can regulate the weight parameters λ_*, λ_1, μ_*, μ_1, λ_q and λ_w to control the imaging performance and mask complexity.

It is observed that the proposed method outperforms the IHT method in image fidelity for both $K = 2$ and $K = 4$. It is noted that the IHT algorithm and the proposed method essentially solve the l_0-norm and l_1-norm CS reconstruction problems, respectively. In CS theory, the l_0-norm is a stronger sparsity regularization than the l_1-norm, but the l_0-norm problem is a combinatorial optimization problem, and cannot be solved efficiently [23]. The IHT method is a greedy algorithm to find out the approximate solution of l_0-norm problem in an efficient way, but the solution is usually suboptimal. On the other hand, it has been proven that the l_0-norm CS reconstruction problem can be equivalently transformed to the l_1-norm reconstruction problem under some constraints [23]. The l_1-norm reconstruction problem is a convex optimization problem that can be efficiently solved by the split Bregman algorithm. From the simulations, we can also observe that the image fidelity obtained by the proposed method with $K = 4$ is inferior than the SD method because of the sparse downsampling on the layout. It is noted that the traditional SD method does not subsample the layout pattern, and optimizes the mask pattern on dense mesh, thus improving the imaging performance at the cost of runtime. Even so, the proposed method with $K = 2$ obtains better image fidelity than the SD method.

Figure 3(a) illustrates the convergence of average pattern errors on both focal and defocus planes corresponding to different ILT methods. It shows that the IHT and proposed methods converge faster than the SD method at the beginning. Compared to the SD method using 150



(a) No. Trapezoids=5 (b) No. Trapezoids=251 (c) No. Trapezoids=241 (d) No. Trapezoids=229 (e) No. Trapezoids=206 (f) No. Trapezoids=191

(g) PE = 10156 EPE = 18.078 nm (h) PE = 3020 EPE = 5.376 nm (i) PE = 2816 EPE = 5.013 nm (j) PE = 4528 EPE = 8.060 nm (k) PE = 2536 EPE = 4.514 nm (l) PE = 3716 EPE = 6.615 nm

(m) PE = 10136 EPE = 18.043 nm (n) PE = 3028 EPE = 5.390 nm (o) PE = 2824 EPE = 5.027 nm (p) PE = 4556 EPE = 8.110 nm (q) PE = 2576 EPE = 4.585 nm (r) PE = 3732 EPE = 6.643 nm

Fig. 2. Simulations of different ILT methods based on vertical line-space pattern.

iterations, the IHT and proposed methods with $K = 2$ can obtain smaller PE values using only 100 iterations. However, the IHT and proposed methods with $K = 4$ converge to larger PE values than the SD method. It is observed that reduction of downsampling rate K will improve the image fidelity of the lithography systems. That is because the small downsampling rate K means more sampling on the target layout. As more compressive measurements are acquired, more information of the original signal will be retained, thus benefiting the improvement of reconstruction accuracy.

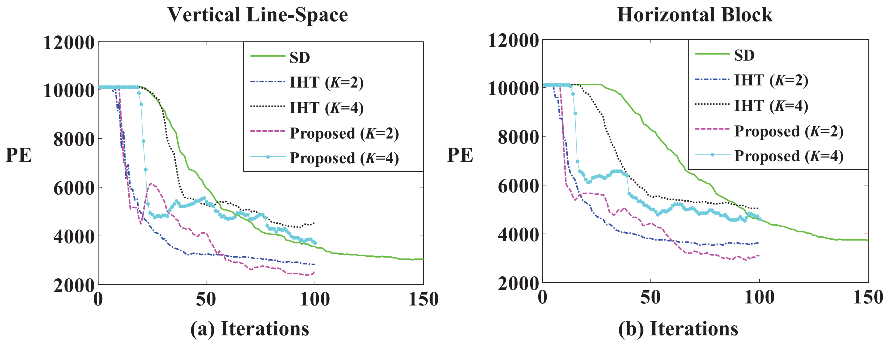


Fig. 3. The convergence of average PEs on both focal and defocus planes obtained by different ILT methods using (a) the vertical line-space pattern and (b) the horizontal block pattern.

Next, we compare the runtimes of different ILT algorithms mentioned above. We calculate the average runtimes over five implementations. For each iteration, the SD method took 614s. The IHT methods with $K = 2$ and $K = 4$ took 359s and 330s, respectively. The proposed methods with $K = 2$ and $K = 4$ took 360s and 329s, respectively. It is shown that the downsampling method could accelerate the speed by 41% to 46% compared to the SD method. In addition, the runtimes of the proposed methods and IHT methods are comparable to each other. Note that the gain in speed is higher than that expected by Eq. (28). It is probably caused by the deviation of runtime in the computer, or by called functions in the optimization codes.

In the following, we evaluate the complexity of mask patterns using the trapezoid count (No. Trapezoids) in the fractured masks [13]. Less number of fractured trapezoids means lower complexity of mask pattern. According to [13], the numbers of fractured trapezoid counts can be approximated calculated as (3/4)(#concave) + (1/4)(#convex), where “#concave” and “#convex” represent the numbers of the concave corners and convex corners in the mask patterns, respectively. The approximate values of trapezoid counts of all mask patterns are presented in Fig. 2. The proposed methods result in lower mask complexity than the SD method and IHT method since the low-rank and sparse penalties in the cost function help to remove singular pixels or details in the mask patterns.

Process window is a commonly used metric for the robustness of lithography system to process variations. Figure 4(a) compares the overlapped process windows obtained by different ILT methods using the vertical line-space pattern, where the *x* and *y* axes represent the depth of focus (DOF) and exposure latitude (EL), respectively. Depth of focus is the maximum deviation of actual imaging plane from the focal plane given an acceptable imaging performance. Exposure latitude represents the tolerance of dose variation to achieve an acceptable imaging performance. The opening of process window encompasses all combinations of DOF and EL satisfying the prescribed metrics of image fidelity. The measurement positions of the process windows are illustrated in Fig. 5(a). In particular, we first calculate the process windows at the locations of ①, ②, ③, and ④ individually. Then, the overlapped area among these four process windows, referred to as the overlapped process window, is shown in Fig. 4(a). The values of DOF corresponding to different ELs are listed in the top half of Table 1. The proposed methods and IHT methods with $K = 2$ and $K = 4$ result in larger or comparable process windows than the SD method. In addition, compared to the IHT methods, the proposed method with $K = 4$ can further improve the DOF at EL = 3%, 5% and 8%.

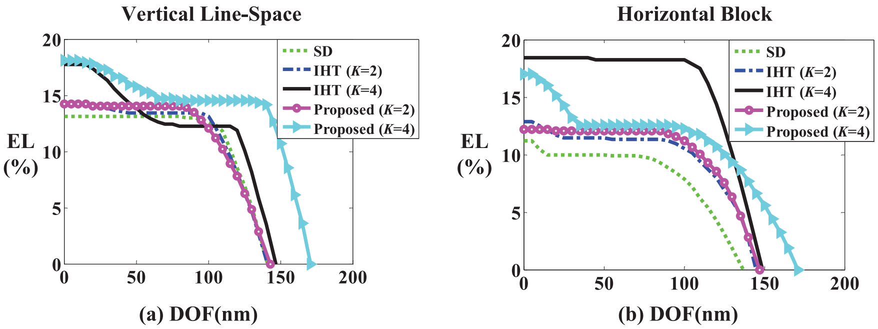


Fig. 4. The overlapped process windows obtained by different ILT methods for (a) the vertical line-space pattern and (b) the horizontal block pattern.

Figure 6 presents the simulations based on another horizontal block pattern at 45nm technology node. The X-polarized illumination is used to enhance the resolution of the horizontally oriented features. Figure 3(b) illustrates the convergence of average PEs on both focal and defocus planes obtained by different ILT methods. Figure 4(b) compares the overlapped process windows, and Fig. 5(b) shows the measurement positions of the process windows. The values of DOF corresponding to different ELs are listed in the bottom half of Table 1. Compared to IHT methods, the proposed methods can further reduce the PE and EPE. In addition, the proposed methods achieve larger or comparable DOF at EL = 3%, 5% and 8% in contrast to the SD and IHT methods. On the other hand, the proposed method leads to lower mask complexity than the SD and IHT methods. Next, we compare the runtimes of different ILT algorithms. For each

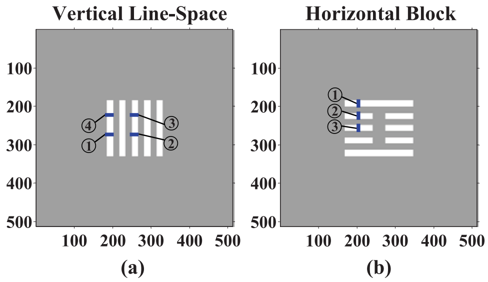


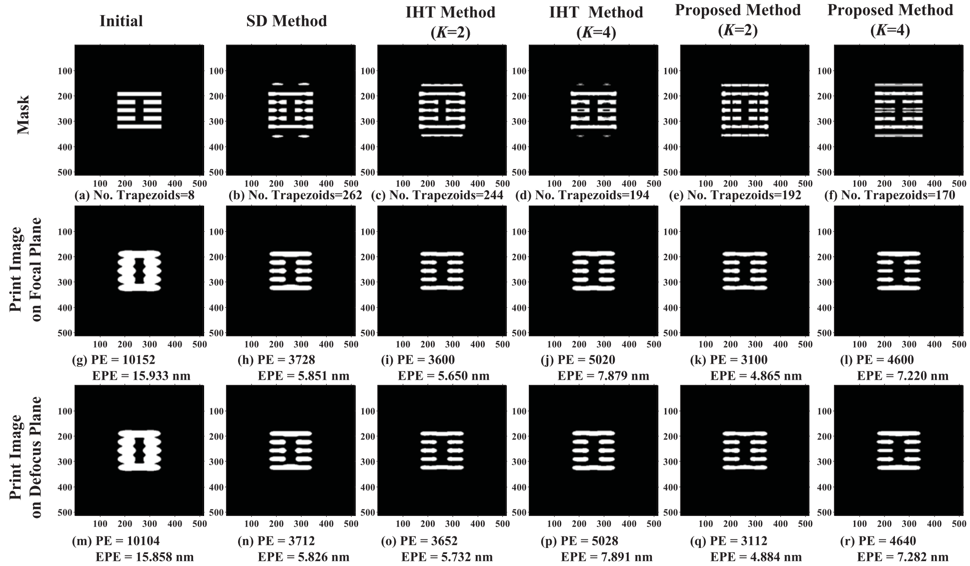
Fig. 5. The measurement positions of the process windows for (a) the vertical line-space pattern and (b) the horizontal block pattern.

Table 1. The values of DOF (nm) corresponding to different ELs.

		Vertical Line-space Pattern			
DOF(nm) at	SD	IHT ($K = 2$)	IHT ($K = 4$)	Proposed ($K = 2$)	Proposed ($K = 4$)
EL = 3%	135	135	140	135	165
EL = 5%	128	128	136	128	162
EL = 8%	122	121	131	118	156
		Horizontal Block Pattern			
DOF(nm) at	SD	IHT ($K = 2$)	IHT ($K = 4$)	Proposed ($K = 2$)	Proposed ($K = 4$)
EL = 3%	126	138	145	138	158
EL = 5%	118	132	139	132	152
EL = 8%	98	120	134	122	138

iteration, the SD method took 613s. The IHT methods with $K = 2$ and $K = 4$ took 358s and 333s, respectively. The proposed methods with $K = 2$ and $K = 4$ took 359s and 332s, respectively. It is shown that the proposed methods result in similar runtimes to the IHT methods, but are more computationally efficient than the SD method.

Figure 7 presents the simulations based on another complex layout pattern and compares the performance of SD method, IHT method ($K = 2$) and the proposed method ($K = 2$). The TE-polarized illumination is used since the layout pattern includes both horizontally and vertically oriented features. For each iteration, the SD method, IHT method and proposed method took 609s, 358s and 332s, respectively. Although the layout pattern in Fig. 7 is more complex than the previous simulations, the dimensions of the mask patterns used in these simulations are the same. Thus, the runtimes for the compelx layout are similar to the runtimes in Figs. 2 and 6. It is shown that the proposed method can further reduce the PE, EPE and mask complexity compared to the SD and IHT methods. In addition, the proposed method can effectively improve the computational efficiency in contrast to the SD method.

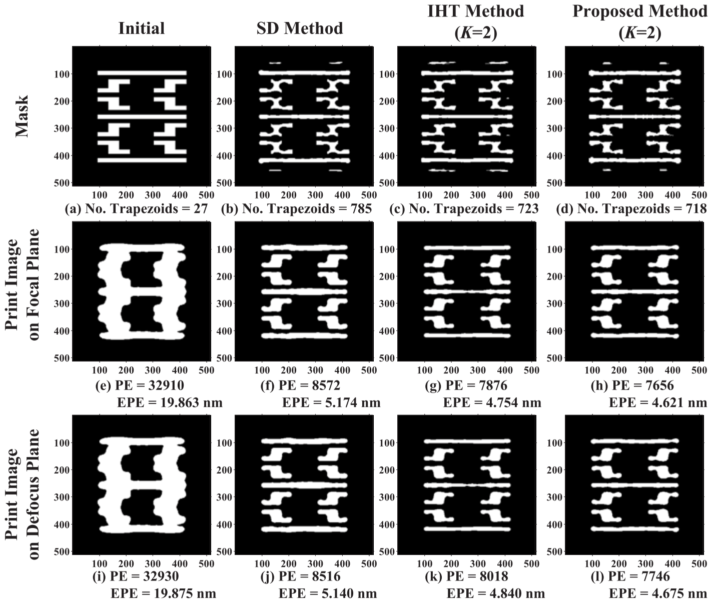


(a) No. Trapezoids=8 (b) No. Trapezoids=262 (c) No. Trapezoids=244 (d) No. Trapezoids=194 (e) No. Trapezoids=192 (f) No. Trapezoids=170

(g) PE = 10152 EPE = 15.933 nm (h) PE = 3728 EPE = 5.851 nm (i) PE = 3600 EPE = 5.650 nm (j) PE = 5020 EPE = 7.879 nm (k) PE = 3100 EPE = 4.865 nm (l) PE = 4600 EPE = 7.220 nm

(m) PE = 10104 EPE = 15.858 nm (n) PE = 3712 EPE = 5.826 nm (o) PE = 3652 EPE = 5.732 nm (p) PE = 5028 EPE = 7.891 nm (q) PE = 3112 EPE = 4.884 nm (r) PE = 4640 EPE = 7.282 nm

Fig. 6. Simulations of different ILT methods based on horizontal block pattern.



(a) No. Trapezoids = 27 (b) No. Trapezoids = 785 (c) No. Trapezoids = 723 (d) No. Trapezoids = 718

(e) PE = 32910 EPE = 19.863 nm (f) PE = 8572 EPE = 5.174 nm (g) PE = 7876 EPE = 4.754 nm (h) PE = 7656 EPE = 4.621 nm

(i) PE = 32930 EPE = 19.875 nm (j) PE = 8516 EPE = 5.140 nm (k) PE = 8018 EPE = 4.840 nm (l) PE = 7746 EPE = 4.675 nm

Fig. 7. Simulations of different ILT methods based on complex layout pattern.

6. Conclusion

This paper developed a fast and robust ILT method based on the nonlinear CS framework. The downsampled layout patterns were used to establish the cost function to effectively reduce the computational complexity. The ILT problem was formulated as a nonlinear inverse optimization problem, where the sparsity and low-rank regularizations were introduced to aid in solving the underdetermined problem and controlling the mask complexity. The split Bregman algorithm was used to solve for the ILT problem and obtain the optimal mask patterns to improve the lithography imaging performance. The computational complexity of the proposed method was analyzed and compared to the traditional SD and IHT methods. Good performance of the proposed methods have been demonstrated by a set of simulations in the aspects of the computational efficiency, imaging performance and mask complexity. In the future work, we will demonstrate the proposed methods using more complex layout patterns at advanced technology nodes.

A. Formula of quadratic penalty and wavelet penalty

According to [8] and [6], the quadratic penalty R_q is formulated as

$$R_q = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\{1-[2\mathbf{M}(i,j)-1]^2\right\} = \mathbf{1}_{N\times1}^T[4\mathbf{M}\odot(\mathbf{1}_{N\times N}-\mathbf{M})]\mathbf{1}_{N\times1}, \tag{29}$$

where $\mathbf{M}(i,j)$ represents the (i,j)th element in the mask pattern, $\mathbf{1}_{N\times1}$ is an $N\times1$ one-valued vector, and $\mathbf{1}_{N\times N}$ is an $N\times N$ one-valued matrix. Based on the parametric transformation in Eq. (8), the quadratic penalty can be rewritten as

$$R_q = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[1-\cos^2\mathbf{\Theta}(i,j)\right] = \mathbf{1}_{N\times1}^T\left(\mathbf{1}_{N\times N}-\cos\mathbf{\Theta}\odot\cos\mathbf{\Theta}\right)\mathbf{1}_{N\times1}, \tag{30}$$

where $\mathbf{\Theta}(i,j)$ represents the (i,j)th element in $\mathbf{\Theta}$.

Next, the derivation of the wavelet penalty is provided. Assume the lateral dimension of mask N is an even number. Given the first-order Haar wavelet transform of the mask pattern $\mathbf{M}$, one can obtain the $(N/2)\times(N/2)$ low-frequency component matrix $\mathbf{A}\in\mathbb{R}^{N/2\times N/2}$, and other three $(N/2)\times(N/2)$ high-frequency component matrices $\mathbf{H}$, $\mathbf{V}$ and $\mathbf{D}\in\mathbb{R}^{N/2\times N/2}$. $\mathbf{H}$, $\mathbf{V}$ and $\mathbf{D}$ represent the high-frequency components in the horizontal, vertical and diagonal directions, respectively. In particular, $\mathbf{A}$, $\mathbf{H}$, $\mathbf{V}$ and $\mathbf{D}$ can be calculated as following

$$\mathbf{A}(i,j) = \mathbf{M}(2i-1,2j-1)+\mathbf{M}(2i-1,2j)+\mathbf{M}(2i,2j-1)+\mathbf{M}(2i,2j), \tag{31}$$

$$\mathbf{H}(i,j) = \mathbf{M}(2i-1,2j-1)-\mathbf{M}(2i-1,2j)+\mathbf{M}(2i,2j-1)-\mathbf{M}(2i,2j), \tag{32}$$

$$\mathbf{V}(i,j) = \mathbf{M}(2i-1,2j-1)+\mathbf{M}(2i-1,2j)-\mathbf{M}(2i,2j-1)-\mathbf{M}(2i,2j), \tag{33}$$

$$\mathbf{D}(i,j) = \mathbf{M}(2i-1,2j-1)-\mathbf{M}(2i-1,2j)-\mathbf{M}(2i,2j-1)+\mathbf{M}(2i,2j), \tag{34}$$

where $i,j = 1,2,\ldots,N/2$. Then, the wavelet penalty R_W is defined as the energy of the high-frequency components included in the mask [6]:

$$R_W = \sum_{i=1}^{N/2}\sum_{j=1}^{N/2}\left[\mathbf{H}(i,j)\times\mathbf{H}(i,j)^*+\mathbf{V}(i,j)\times\mathbf{V}(i,j)^*+\mathbf{D}(i,j)\times\mathbf{D}(i,j)^*\right]. \tag{35}$$

Based on the parametric transformation in Eq. (8), the wavelet penalty can be rewritten as a function of $\mathbf{\Theta}$. If N is an odd number, we could ignore the bottom row and the rightmost column in the mask pattern.

B. Derivation of cost function gradient

In Eq. (16), we define a new cost function G as

$$\begin{aligned} G(\boldsymbol{\Theta}) = {} & \alpha T_K + (1-\alpha)T'_K + \mu_* \|\mathbf{D} - \boldsymbol{\Theta} - \mathbf{V}\|_2^2 + \mu_1 \|\mathbf{C} - \bar{\boldsymbol{\Theta}} - \mathbf{W}\|_2^2 \\ & + \lambda_q R_q(\boldsymbol{\Theta}) + \lambda_w R_w(\boldsymbol{\Theta}), \end{aligned} \tag{36}$$

where $T_K = \|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}_K(\boldsymbol{\Theta}) + \mathbf{E}_K]\|_2^2$, and $T'_K = \|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}'_K(\boldsymbol{\Theta}) + \mathbf{E}'_K]\|_2^2$. The gradient of the cost function with respect to $\boldsymbol{\Theta}$ is calculated as

$$\begin{aligned} \nabla G(\boldsymbol{\Theta}) = {} & \alpha \nabla T_K(\boldsymbol{\Theta}) + (1-\alpha)\nabla T'_K(\boldsymbol{\Theta}) + 2\mu_*(\boldsymbol{\Theta} + \mathbf{V} - \mathbf{D}) + 2\mu_1 \boldsymbol{\Psi}^T(\bar{\boldsymbol{\Theta}} + \mathbf{W} - \mathbf{C})\boldsymbol{\Psi} \\ & + \lambda_q \nabla R_q(\boldsymbol{\Theta}) + \lambda_w \nabla R_w(\boldsymbol{\Theta}), \end{aligned} \tag{37}$$

where $\nabla T_K(\boldsymbol{\Theta})$ and $\nabla T'_K(\boldsymbol{\Theta})$ are the gradients of T_K and T'_K, respectively. Next, we will provide the detailed method to calculate $\nabla T_K(\boldsymbol{\Theta})$, and $\nabla T'_K(\boldsymbol{\Theta})$ can be calculated in the same manner.

Note that T_K can be expressed as

$$\begin{aligned} T_K &= \|\boldsymbol{\Pi}_K \odot [\tilde{\mathbf{Z}}_K - \mathbf{Z}_K(\boldsymbol{\Theta}) + \mathbf{E}_K]\|_2^2 \\ &= \sum_{m=1}^{N/K}\sum_{n=1}^{N/K} \left\{\boldsymbol{\Pi}(Km,Kn) \times \left[\tilde{\mathbf{Z}}(Km,Kn) - \mathbf{Z}(Km,Kn) + \mathbf{E}_K(Km,Kn)\right]\right\}^2 . \end{aligned} \tag{38}$$

According to the lithography imaging model and photoresist model, the partial derivative of T_K to the mask variable $\mathbf{M}(r,s)$ can be calculated as

$$\begin{aligned} \frac{\partial T_K}{\partial \mathbf{M}(r,s)} = {} & -\frac{4a}{J_{\text{sum}}} \sum_{x_s}\sum_{y_s} \mathbf{J}(x_s,y_s) \times \sum_{p=x,y,z} \text{Re}\left([\mathbf{B}^{x_s y_s}(r,s)]^*\right. \\ & \times \sum_{m=1}^{N/K}\sum_{n=1}^{N/K} \left[\mathbf{H}_p^{x_s y_s}(Km-r,Kn-s)\right]^* \times \{\boldsymbol{\Pi}(Km,Kn) \times \mathbf{Z}(Km,Kn) \\ & \times \left[\tilde{\mathbf{Z}}(Km,Kn) - \mathbf{Z}(Km,Kn) + \mathbf{E}_K(Km,Kn)\right] \times [1 - \mathbf{Z}(Km,Kn)]\} \\ & \left.\times \left\{\sum_r\sum_s \mathbf{H}_p^{x_s y_s}(Km-r,Kn-s) \times [\mathbf{B}^{x_s y_s}(r,s) \times \mathbf{M}(r,s)]\right\}\right), \end{aligned} \tag{39}$$

where the notation $*$ is the conjugate operation, and $\text{Re}(\cdot)$ represents the real part of the argument. Make parameter transformations $r = Ka + u$ and $s = Kb + v$, and replace $\mathbf{H}_p^{x_s y_s}(Kx-u,Ky-v)$, $\mathbf{B}^{x_s y_s}(Kx+u,Ky+v)$, and $\mathbf{M}(Kx+u,Ky+v)$ by $\mathbf{H}_{p,uv}^{x_s y_s}(x,y)$, $\mathbf{B}_{uv}^{x_s y_s}(x,y)$ and $\mathbf{M}_{uv}(x,y)$, respectively. Then, we have

$$\begin{aligned} \frac{\partial T_K}{\partial \mathbf{M}_{uv}(a,b)} = {} & -\frac{4a}{J_{\text{sum}}} \sum_{x_s}\sum_{y_s} \mathbf{J}(x_s,y_s) \times \sum_{p=x,y,z} \text{Re}\left(\left[\mathbf{B}_{uv}^{x_s y_s}(a,b)\right]^*\right. \\ & \times \sum_{m=1}^{N/K}\sum_{n=1}^{N/K} \left[\mathbf{H}_{p,uv}^{x_s y_s}(m-a,n-b)\right]^* \times \{\boldsymbol{\Pi}(Km,Kn) \times \mathbf{Z}(Km,Kn) \\ & \times \left[\tilde{\mathbf{Z}}(Km,Kn) - \mathbf{Z}(Km,Kn) + \mathbf{E}_K(Km,Kn)\right] \times [1 - \mathbf{Z}(Km,Kn)]\} \\ & \left.\times \left[\sum_{a=0}^{N/K-1}\sum_{b=0}^{N/K-1} \mathbf{B}_{uv}^{x_s y_s}(a,b) \times \mathbf{H}_{p,uv}^{x_s y_s}(m-a,n-b) \times \mathbf{M}_{uv}(a,b)\right]\right]\right). \end{aligned} \tag{40}$$

Thus, the gradient of T_K with respect to $\mathbf{M}_{uv}$ can be calculated as

$$\begin{aligned} \nabla T_{K,uv} = {} & -\frac{4a}{J_{\text{sum}}} \sum_{x_s}\sum_{y_s} \mathbf{J}(x_s,y_s) \times \sum_{p=x,y,z} \text{Re}\left(\left(\mathbf{B}_{uv}^{x_s y_s}\right)^* \odot \left(\mathbf{H}_{p,uv}^{x_s y_s}\right)^{*\circ} \otimes \left\{\left[\mathbf{H}_{p,uv}^{x_s y_s}\right.\right.\right. \\ & \left.\left.\left.\otimes \left(\mathbf{B}_{uv}^{x_s y_s} \odot \mathbf{M}_{uv}\right)\right] \odot \boldsymbol{\Pi}_K \odot \left(\tilde{\mathbf{Z}}_K - \mathbf{Z}_K + \mathbf{E}_K\right) \odot \mathbf{Z}_K \odot (\mathbf{1}_K - \mathbf{Z}_K)\right\}\right), \end{aligned} \tag{41}$$

where $\nabla T_{K,uv} \in \mathbb{R}^{N/K \times N/K}$, the notation $\circ$ is the operator to rotate the matrix by 180° in both horizontal and vertical directions, and $\mathbf{1}_K$ is an $N/K \times N/K$ one-valued matrix. Note that ∇T_K is an $N \times N$ matrix, and $\nabla T_{K,uv}$ in Eq. (41) is the downsampling version of ∇T_K. Thus, the elements in ∇T_K can be calculated as following:

$$\nabla T_K(Ka + u, Kb + v) = \nabla T_{K,uv}(a, b), \tag{42}$$

where $u, v = 1, \ldots, K$, and $a, b = 0, \ldots, N/K - 1$. Using the Chain rule, the partial derivative of T_K to $\mathbf{\Theta}(r, s)$ is given by

$$\frac{\partial T_K}{\partial \mathbf{\Theta}(r,s)} = \frac{\partial T_K}{\partial \mathbf{M}(r,s)} \times \frac{\partial \mathbf{M}(r,s)}{\partial \mathbf{\Theta}(r,s)}. \tag{43}$$

According to Eq. (8), the gradient of T_K with respect to $\mathbf{\Theta}$ is

$$\nabla T_K(\mathbf{\Theta}) = -\frac{1}{2} \nabla T_K \odot \sin\mathbf{\Theta}. \tag{44}$$

Using the same method, we can calculate the gradient $\nabla T'_K$.

In Eq. (37), $\nabla R_q(\mathbf{\Theta})$ and $\nabla R_w(\mathbf{\Theta})$ are the gradients of R_q and R_w with respect to $\mathbf{\Theta}$, respectively. Due to the length limit of this paper, we skip the method to calculate $\nabla R_q(\mathbf{\Theta})$ and $\nabla R_w(\mathbf{\Theta})$, which are provided in [6] and [8].

Funding

National Natural Science Foundation of China (61675021); Natural Science Foundation of Beijing Municipality (4173078); Fundamental Research Funds for the Central Universities (2018CX01025); China Scholarship Council (201706035012).

References

1. F. Schellenberg, "A little light magic," IEEE Spectrum **40**(9), 34–39 (2003).
2. Y. Wei, *Advanced Lithography Theory and Application of VLSI* (Science Press, 2016).
3. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography* (SPIE, 2001).
4. X. Ma and G. R. Arce, *Computational Lithography*, Wiley Series in Pure and Applied Optics, 1st ed. (John Wiley and Sons, 2010).
5. C. Chu, B. Tsaoa, K. Chiou, S. Lee, J. Huang, Y. Liu, T. Lin, A. Moore, and L. Pang, "Enhancing DRAM printing process window by using inverse lithography technology (ILT)," Proc. SPIE **6154**, 61543O (2006).
6. X. Ma, Y. Li, and L. Dong, "Mask optimization approaches in optical lithography based on a vector imaging model," J. Opt. Soc. Am. A **29**(7), 1300–1312 (2012).
7. Y. Granik, "Fast pixel-based mask optimization for inverse lithography," J. Micro/Nanolithogr., MEMS, MOEMS **5**(4), 043002 (2006).
8. A. Poonawala and P. Milanfar, "Mask design for optical microlithography-an inverse imaging problem," IEEE Trans. Image Processing **16**(3), 774–788 (2007).
9. X. Ma and G. R. Arce, "Pixel-based OPC optimization based on conjugate gradients," Opt. Express **19**(3), 2165–2180 (2011).
10. P. M. Martin, C. J. Progler, G. Xiao, R. Gray, L. Pang, and Y. Liu, "Manufacturability study of masks created by inverse lithography technology (ILT)," Proc. SPIE **5992**, 599235 (2005).
11. B. Kim, S. S. Suh, S. G. Woo, H. Cho, G. Xiao, D. H. Son, D. Irby, D. Kim, and K. Baik, "Inverse lithography technology (ILT) mask manufacturability for full-chip device," Proc. SPIE **7488**, 748812 (2009).
12. X. Ma, Z. Song, Y. Li, and G. R. Arce, "Block-based mask optimization for optical lithography," Appl. Opt. **52**(14), 3351–3363 (2013).
13. X. Ma and Y. Li, "Resolution enhancement optimization methods in optical lithography with improved manufacturability," J. Micro/Nanolithogr., MEMS, MOEMS **10**(2), 023009 (2011).
14. X Ma, S. Jiang, and A. Zakhor, "A cost-driven fracture heuristics to minimize external sliver length," Proc. SPIE **7973**, 79732O (2011).
15. Y. Liu and A. Zakhor, "Binary and phase shifting mask design for optical lithography," IEEE Trans. Semiconduct. Manufact. **5**(2), 138–152 (1992).
16. N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," J. Opt. **12**(4), 45601–45609 (2010).
17. Y. Shen, N. Jia, N. Wong, and E. Y. Lam, "Robust levelset-based inverse lithography," Opt. Express **19**(6), 5511–5521 (2011).

18. J. Yu and P. Yu, "Impacts of cost functions on inverse lithography patterning," Opt. Express **18**(22), 23331–23342 (2010).
19. W. Lv, S. Liu, Q. Xia, X. Wu, Y. Shen, and E. Y. Lam, "Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step," J. Vac. Sci. Technol., B **31**(4), 041605 (2013).
20. X. Ma, D. Shi, Z. Wang, Y. Li, and G. R. Arce, "Lithographic source optimization based on adaptive projection compressive sensing," Opt. Express **25**(6), 7131–7149 (2017).
21. X. Ma, Z. Wang, H. Lin, Y. Li, G. R. Arce, and L. Zhang, "Optimization of lithography source illumination arrays using diffraction subspaces," Opt. Express **26**(4), 3738–3755 (2018).
22. D. Donoho, "Compressive sensing," IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006).
23. E. Candés, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Trans. Inf. Theory **52**(2), 489–509 (2006).
24. S. Yang, M. Wang, P. Li, L. Jin, B. Wu, and L. Jiao, "Compressive hyperspectral imaging via sparse tensor and nonlinear compressed sensing," IEEE Trans. Geosci. Electron. **53**(11), 5943–5957 (2015).
25. H. Chen, H. Kung, and M. Comiter, "Nonlinear compressive sensing for distorted measurements and application to improving efficiency of power amplifiers,“ *IEEE International Conference on Communications*, 1–7 (2017).
26. J. Ke and E. Y. Lam, Nonlinear image reconstruction in block-based compressive imaging, *IEEE International Symposium on Circuits and Systems*, 2917–2920 (2012).
27. T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," Appl. Comput. Harmon. A. **27**(3), 265–274 (2009).
28. T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," J. Fourier Anal. Appl. **14**(5-6), 629–654 (2008).
29. T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimization problems," IEEE Trans. Inf. Theory **59**(6), 3466–3474 (2013).
30. X. Ma, Z. Wang, Y. Li, G. R. Arce, L. Dong, and J. G. Frias, "Fast optical proximity correction method based on nonlinear compressive sensing," Opt. Express **26**(11), 14479–14498 (2018).
31. H. Gao, H. Y. Yu, S. Osher, and G. Wang, "Multi-energy CT based on a prior rank, intensity and sparsity model (PRISM)," Inverse Probl. **27**(11), 115012 (2011).
32. L. Li, Z. Chen, G. Wang, J. Chu, and H. Gao, "A tensor PRISM algorithm for multi-energy CT reconstruction and comparative studies," J. X-Ray Sci. Technol. **22**(2), 147–163 (2014).
33. V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," arXiv:0906.2220v1, Jun. 2009.
34. T. Goldstein and S. Osher, "The split Bregman method for l_1-regularized problems," SIAM J. Imaging Sci. **2**(2), 323–343 (2009).
35. D. Peng, P. Hu, V. Tolani, and T. Dam, "Toward a consistent and accurate approach to modeling projection optics," Proc. SPIE **7640**, 76402Y (2010).
36. X. Ma, C. Han, Y. Li, L. Dong, and G. R. Arce, "Pixelated source and mask optimization for immersion lithography," J. Opt. Soc. Am. A **30**(1), 112–123 (2013).
37. J. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," Multiscale Model. Simul. **8**(2), 337–369 (2010).
38. J. F. Cai, E. J. Candés, and Z. Shen, "A singular value thresholding algorithm for matrix completion," SIAM J. Control **20**(4), 1956–1982 (2010).