PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Comparison of different lithographic source optimization methods based on compressive sensing

Wang, Zhiqiang, Ma, Xu, Chen, Rui, Arce, Gonzalo, Dong, Lisong, et al.

Zhiqiang Wang, Xu Ma, Rui Chen, Gonzalo R. Arce, Lisong Dong, Hans-Juergen Stock, Yayi Wei, "Comparison of different lithographic source optimization methods based on compressive sensing," Proc. SPIE 11327, Optical Microlithography XXXIII, 1132716 (23 March 2020); doi: 10.1117/12.2551037

SPIE.

Event: SPIE Advanced Lithography, 2020, San Jose, California, United States

Comparison of Different Lithographic Source Optimization Methods Based on Compressive Sensing

Zhiqiang Wang[a], Xu Ma[a*], Rui Chen[b*], Gonzalo R. Arce[c], Lisong Dong[b], Hans-Juergen Stock[d], and Yayi Wei[b]

[a]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

[b]Integrated Circuit Advanced Process Center, Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China

[c]Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, 19716, USA

[d]Synopsys GmbH, Karl-Hammerschmidt-Strasse 34, 85609 Aschheim/Dornach, Germany.

ABSTRACT

Source optimization (SO) is a widely used resolution enhancement technique to improve the imaging performance of optical lithography systems. Recently, a fast pixelated SO method for inverse lithography has been developed based on the theory of compressive sensing (CS). In last several years, CS has explored numerous reconstruction algorithms to solve for inverse problems. These algorithms are critical in attaining good reconstruction quality also aiming at reducing the time complexity. This paper compares different SO methods based on CS algorithms including the linearized Bregman (LB) algorithm, the alternating direction method of multipliers (ADMM), the fast iterative shrinkage-thresholding algorithm (FISTA), the approximate message-passing (AMP), and the gradient projection for sparse reconstruction (GPSR). Benefiting from the strategy of variable splitting and adaptive step size searching, the GPSR method effectively retains the optimization efficiency. Computational experiments also show that the GPSR method can achieve superior or comparable SO performance on average over other methods. It is also shown that the proposed SO methods can be applied to develop a fast source-mask optimization (SMO) method based on the CS framework.

Keywords: optical lithography, computational lithography, source optimization (SO), compressive sensing (CS), source-mask optimization (SMO)

1. INTRODUCTION

Optical lithography is a fundamental method to fabricate integrated micro devices on a semiconductor chip through a series of physical and chemical processes. Nowadays, resolution enhancement techniques (RETs) have become indispensable to improve the imaging resolution and fidelity in the optical lithography systems, especially with the scaling down of the critical dimension (CD) in integrated circuits.[1] Source optimization (SO) is regarded as an important RET to achieve better imaging performance, which aims at modulating the intensities and directions of the incident light rays in order to compensate for the image distortion.[2] In the past, several SO methods have been proposed to design and optimize the structured illumination in lithography systems. Brist, et al. used the parametric methods to optimize the source patterns with similar geometries as the traditional off-axis illuminations (OAI).[3] Burkhardt, et al. proposed a source design algorithm, increasing the contrast for arbitrary periodic mask features by removing certain parts of the circular illumination.[4] Gau, et al. designed a customized illumination aperture filter to optimize the source for features at many pitches.[5] Subsequently, with the advances of programmable lithography illuminations,[6] pixelated SO methods were proposed to improve the

Further author information: (Send correspondence to X.M. and R.C.)
X.M.: E-mail: maxu@bit.edu.cn, Telephone: +861068912256
R.C.: E-mail: chenrui@ime.ac.cn, Telephone: +861082995779

Optical Microlithography XXXIII, edited by Soichi Owa, Proc. of SPIE Vol. 11327, 1132716
 · doi: 10.1117/12.2551037

imaging performance attributed to the high degree of optimization freedom. Pixelated SO modifies the intensities of all pixels on the source pattern. Recently, a branch of mathematical and algorithmic approaches, termed as computational lithography have been proposed to solved for the inverse SO problems, and the advantages of pixelated SO methods have also been proved.[7–11] In addition, the pixelated SO methods can be also applied to develop more advanced computational lithography methods by jointly optimizing the lithographic source, mask, illumination polarization, pupil wavefront and so on.[12–16]

However, the pixelated SO methods suffer from intensive computational complexity because of the large number of optimization variables, especially for very-large and ultra-large scale semiconductor fabrication. Inspired by the compressive sensing (CS) theory,[17,18] a set of fast pixelated SO approaches were recently developed based on the linear CS frameworks,[19,20] where the optimal source pattern is derived from the following inverse reconstruction problem:

$$\begin{aligned} \hat{\vec{\theta}} &= \underset{\vec{\theta}}{\operatorname{argmin}} \|\vec{\theta}\|_1 \quad \text{s.t.} \quad \mathbf{\Phi}\vec{Z}_s = \mathbf{\Phi}\mathbf{I}_{cc}^s\vec{J} \\ &= \underset{\vec{\theta}}{\operatorname{argmin}} \|\vec{\theta}\|_1 \quad \text{s.t.} \quad \mathbf{\Phi}\vec{Z}_s = \mathbf{\Phi}\mathbf{I}_{cc}^s\mathbf{\Psi}\vec{\theta}, \end{aligned} \tag{1}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the l_1-norm and l_2-norm, respectively. $\vec{Z}_s \in \mathbb{R}^{M\times 1}$ is the vectorized representation of the target layout supported by the M ($M \ll N_s^2$) monitoring pixels sparsely distributed on the layout pattern, where N_s is the lateral dimension of the source pattern. $\mathbf{I}_{cc}^s \in \mathbb{R}^{M\times N_s^2}$ is the illumination cross coefficient (ICC) matrix corresponding to the selected M monitoring pixels. $\vec{J} \in \mathbb{R}^{N_s^2\times 1}$ is the raster-scanned vector of the source pattern that can be sparsely represented on a predefined basis, that is $\vec{J} = \mathbf{\Psi}\vec{\theta}$, where $\mathbf{\Psi} \in \mathbb{R}^{N_s^2\times N_s^2}$ and $\vec{\theta} \in \mathbb{R}^{N_s^2\times 1}$ denote the sparse basis and the sparse coefficients, respectively. $\mathbf{\Phi} \in \mathbb{R}^{L\times M}$ ($L < M$) is an adaptive or random projection matrix to further reduce the dimensionality of the linear equations in Eq. (1). In the above problem, the linear constraints are used to improve the image fidelity of the lithography system, while the l_1-norm minimization is used to reduce the solution space for the underdetermined problem. Note that the dimensionality of the linear equations in Eq. (1) is much less than the source variables, and then the computational efficiency can be effectively improved.

In the past, the linearized Bregman (LB) algorithm has been exploited to solve for the SO problem based on the CS frameworks (CS-SO problem for short).[20–22] It proves that the CS-SO algorithms are much faster than the conventional gradient-based algorithm, while retaining better lithographic imaging performance.[19,20] It is noted that the physically manufacturable source pattern is nonnegative, since the value of every source pixel indicates the light intensity emitted from that source point. Therefore, it is necessary to keep the values of the source pixels greater than or equal to zero in each iteration. However, the LB algorithm needs to appropriately choose a positive hard threshold value to keep the nonnegativity of the source pattern, and uses a constant step size to update the source variables. But, the convergence performance of the LB algorithm is sensitive to the hard threshold value and the step size.

This paper will use the gradient projection for sparse reconstruction (GPSR) algorithm to solve for the CS-SO problem. The variable splitting strategy is used to keep the source intensity to be nonnegative. Meanwhile, the GPSR algorithm can produce preferable step sizes using an adaptive method,[23] which is beneficial to improve the computational speed and the convergence property. For comparison, other CS reconstruction algorithms are utilized to solve for the CS-SO problem, including the alternating direction method of multipliers (ADMM),[24] the fast iterative shrinkage-thresholding algorithm (FISTA),[25] the approximate message-passing (AMP) algorithm,[26] and the LB algorithm as mentioned above. It will be shown that the GPSR algorithm can achieve better or comparable lithography imaging performance compared to other algorithms, and retain the optimization efficiency. It is noted that the AMP algorithm is faster than the GPSR algorithm, but may lead to a worse imaging performance. Furthermore, a fast pixelated source-mask optimization (SMO) method is developed by combining the proposed CS-SO method with the fast optical proximity correction (OPC) method based on the nonlinear CS framework.[27] The SMO based on CS method (CS-SMO for short) provides an improvement over the traditional gradient-based SMO method in terms of computational efficiency and imaging performance.

In the following, the fundamentals of GPSR algorithm are provided in Section 2. The SO framework based on

GPSR algorithm is proposed in Section 3. Simulations and comparisons are presented in Section 4. Conclusions are provided in Section 5.

2. FUNDAMENTALS OF GPSR ALGORITHM

Suppose a signal $\vec{x} \in \mathbb{R}^{N\times 1}$ is sparse on a predefined basis, and can be represented as $\vec{x} = \mathbf{\Psi}\vec{\theta}$, where $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$ is the sparse basis, and $\vec{\theta} \in \mathbb{R}^{N\times 1}$ is the sparse coefficient vector, in which only K ($K \ll N$) elements are nonzero or significant. CS theory shows that the sparse signal $\vec{x}$ can be recovered from a small set of compressive measurement $\vec{y} \in \mathbb{R}^{M\times 1}$ ($K < M \ll N$),[17,18] where $\vec{y} = \mathbf{\Phi}\vec{x} = \mathbf{\Phi}\mathbf{\Psi}\vec{\theta}$ and $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ is the projection matrix. The GPRS algorithm can be used to reconstruct the signal $\vec{x}$ by solving for the vector $\vec{\theta}$ from the following optimization problem:

$$\hat{\vec{\theta}} = \min_{\vec{\theta}} \frac{1}{2}\|\vec{y} - \mathbf{\Phi}\mathbf{\Psi}\vec{\theta}\|_2^2 + \tau\|\vec{\theta}\|_1, \tag{2}$$

where $\mathbf{A} = \mathbf{\Phi}\mathbf{\Psi}$ is known as the sensing matrix, and τ is a positive weight parameter.

In GPSR algorithm, the variable $\vec{\theta}$ is split into the positive part and the negative part. Thus, we introduce the vectors $\vec{u}$ and $\vec{v}$, and make the substitution $\vec{\theta} = \vec{u} - \vec{v}$, where all elements of $\vec{u}$ and $\vec{v}$ are positive. It is worth indicating that if the solution of Eq. (2) is prejudged to be nonnegative, the negative component $\vec{v}$ should be ignored, thus $\vec{\theta} = \vec{u}$. Therefore, Eq. (2) can be rewritten into the following bound-constrained quadratic program form:

$$\min_{\vec{u}} \ (\tau\mathbf{1}_N - \mathbf{A}^T\vec{y})^T\vec{u} + \frac{1}{2}\vec{u}^T\mathbf{A}^T\mathbf{A}\vec{u} \equiv F(\vec{u}), \quad \text{s.t.} \ \ \vec{u} \geqslant 0, \tag{3}$$

where $\mathbf{1}_N$ is a $1 \times N$ vector of ones. The GPSR algorithm iteratively updates the signal to pursue the optimal solution of Eq. (3). Let $\vec{u}^n$ represent the estimate of the coefficient vector in the nth iteration. Then, in the $(n+1)$th iteration we could update the coefficient vector as

$$\vec{u}^{n+1} = \left[\vec{u}^n - \alpha^n \nabla F(\vec{u}^n)\right]_+, \tag{4}$$

where $[\cdot]_+$ denotes the operator to extract the positive part of the argument, which is defined as $[x]_+ = max(0, x)$. $\nabla F(\vec{u}^n)$ is the gradient of the objective function with respect to $\vec{u}^n$, which is given by $\nabla F(\vec{u}^n) = \tau\mathbf{1}_N^T - \mathbf{A}^T\vec{y} + 1/2\mathbf{A}^T\mathbf{A}\vec{u}^n$. The step size α^n in Eq. (4) is selected as the first number in the sequence $\alpha_0, \beta\alpha_0, \beta^2\alpha_0, ...$ that satisfies

$$F\left\{\left[\vec{u}^n - \alpha^n\nabla F(\vec{u}^n)\right]_+\right\} \leq F(\vec{u}^n) - \gamma\nabla F(\vec{u}^n)^T\left[\vec{u}^n - \alpha^n\nabla F(\vec{u}^n)\right]_+, \tag{5}$$

where $\beta \in (0,1)$, $\gamma \in (0, 1/2)$, $\alpha_0 = [(g^n)^T g^n]/[(g^n)^T\mathbf{A}^T\mathbf{A}g^n]$, and g^n is an $N \times 1$ vector with its ith element given by

$$g_i^n = \begin{cases} \nabla F(\vec{u}^n)_i, & \text{if } \vec{u}_i^n > 0 \text{ or } \nabla F(\vec{u}^n)_i < 0 \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

In the above equation, $\vec{u}_i^n$ and $F(\vec{u}^n)_i$ are the ith elements of $\vec{u}^n$ and $F(\vec{u}^n)$, respectively. Note that in Eq. (4) we do not need to select the appropriate positive threshold value, since the operation $[\ \cdot\ ]_+$ will automatically keep the variables nonnegative. Meanwhile, the GPSR algorithm can produce a preferable step size based on the method in Eqs. (5) and (6), thus improving the computational speed and the convergence property.

3. SO FRAMEWORK BASED ON GPSR ALGORITHM

Let $\mathbf{J} \in \mathbb{R}^{N_s\times N_s}$ and $\mathbf{M} \in \mathbb{R}^{N\times N}$ be the source pattern and mask pattern, respectively. According to the Abbe's imaging model,[28,29] the aerial image $\mathbf{I}$ of the lithography system can be calculated as:

$$\mathbf{I} = \frac{1}{J_{\text{sum}}}\sum_{x_s}\sum_{y_s}\left[\mathbf{J}(x_s, y_s) \times \sum_{p=x,y,z}|\mathbf{H}_p^{x_s y_s} \otimes (\mathbf{B}^{x_s y_s} \odot \mathbf{M})|^2\right], \tag{7}$$

where (x_s, y_s) represents the spatial coordinate on the source plane, $\mathbf{J}(x_s, y_s)$ is the intensity at the source point (x_s, y_s), and $J_{sum} = \sum_{x_s}\sum_{y_s}\mathbf{J}(x_s, y_s)$ is a normalization factor. $\mathbf{B}^{x_s y_s}$ indicates the phase shift induced by

the oblique incidence of the light rays, $\mathbf{H}_p^{x_s y_s}$ is the point spread function, and the notations $\otimes$ and $\odot$ represent the convolution operator and the element-wise multiplication, respectively. Suppose $\vec{J} \in \mathbb{R}^{N_s^2 \times 1}$ and $\vec{I} \in \mathbb{R}^{N^2 \times 1}$ are the raster-scanned vectors of the source pattern $\mathbf{J}$ and the aerial image $\mathbf{I}$, respectively. Equation (7) can be converted into the following form:

$$\vec{I} = \mathbf{I}_{cc}\vec{J}, \tag{8}$$

where $\mathbf{I}_{cc} \in \mathbb{R}^{N^2 \times N_s^2}$ is the ICC matrix that maps the source pattern to the resulting aerial image.

In the ideal case, the actual aerial image should be approximately equal to the desired pattern, i.e., $\vec{Z} = \vec{I} = \mathbf{I}_{cc}\vec{J}$, where $\vec{Z} \in \mathbb{R}^{N^2 \times 1}$ is the vectorized representation of the target layout. In order to compress the dimensionality of the linear equations, we select $M (M \ll N_s^2 \ll N^2)$ monitoring pixels on the layout pattern using the blue noise sampling method.[20] Then, we only preserve the linear equations supported by those M monitoring pixels. In addition, the adaptive projection matrix $\mathbf{\Phi}$ is used to further reduce the dimensionality of the linear equations. The design method of the adaptive projection matrix can be found in Ref. [20]. Then, the SO problem is finally formulated as the following format:

$$\begin{aligned} \hat{\vec{\theta}} &= \underset{\vec{\theta}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{\Phi}\vec{Z}_s - \mathbf{\Phi}\mathbf{I}_{cc}^s\vec{J}\|_2^2 + \tau\|\vec{\theta}\|_1, \quad \text{s.t.} \;\; \vec{J} \geq 0 \\ &= \underset{\vec{\theta}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{\Phi}\vec{Z}_s - \mathbf{\Phi}\mathbf{I}_{cc}^s\mathbf{\Psi}\vec{\theta}\|_2^2 + \tau\|\vec{\theta}\|_1, \quad \text{s.t.} \;\; \vec{\theta} \geq 0, \end{aligned} \tag{9}$$

where $\mathbf{\Psi}$ is the Gaussian basis matrix to sparsely represent the source pattern,[19] and $\mathbf{I}_{cc}^s \in \mathbb{R}^{M \times N_s^2}$ and $\vec{Z}^s \in \mathbb{R}^{M \times 1}$ represent the ICC matrix and the target layout vector corresponding to the M monitoring pixels. Since all elements in the Gaussian basis $\mathbf{\Psi}$ are nonnegative, the condition $\vec{\theta} \geq 0$ will lead to $\vec{J} = \mathbf{\Psi}\vec{\theta} \geq 0$. According to Eq. (3), the problem in Eq. (9) can be transformed into the following form:

$$\min_{\vec{\theta}} \; (\tau \mathbf{1}_{N_s^2} - \mathbf{B}^T\vec{q})^T\vec{\theta} + \frac{1}{2}\vec{\theta}^T\mathbf{B}^T\mathbf{B}\vec{\theta} \equiv F(\vec{\theta}), \quad \text{s.t.} \;\; \vec{\theta} \geqslant 0, \tag{10}$$

where $\mathbf{B} = \mathbf{\Phi}\mathbf{I}_{cc}^s\mathbf{\Psi}$, $\vec{q} = \mathbf{\Phi}\vec{Z}_s$. Then, Eq. (9) can be solved using the GPSR algorithm described in Section 2.

4. SIMULATION AND COMPARISON

4.1 SO simulation results using different CS algorithms

This section provides the simulation results of the CS-SO method based on the GPSR algorithm. Besides, different CS algorithms are compared in solving the CS-SO problem. The simulations are based on a 193nm ArF immersion lithography system. The numerical aperture (NA) on the wafer side is 1.2, the refractive index of the immersion liquid between the projector and the wafer is 1.44, and the demagnification factor of the projection system is 4. Figure 1 illustrates the simulations based on a line-space layout pattern with CD=45nm. The first row shows the results of optimized sources using different CS algorithms. The second row illustrates the print images corresponding to the source patterns in the first row. From left to right, it shows the simulation results obtained by the FISTA, AMP, ADMM, LB and GPSR algorithms, respectively. It also provides the values of pattern error (PE) and edge placement error (EPE) of the print images, as well as the normalized image log slop (NILS) and contrast of the aerial images. The PE is defined as the square of the Euclidean distance between the target layout and the print image. The EPE indicates the error of the actual printed edge position with respect to the target. The contrast and NILS are calculated along the red dashed lines on the aerial images. The NILS measures the slope of the aerial image on the image contour. It is observed that the imaging performance obtained by the GPSR algorithm is better than other algorithms. In addition, Table 1 lists the iteration number, the total runtime and the runtime per iteration for different CS algorithms. It shows that the GPSR algorithm is faster than the other algorithms except for the AMP algorithm. It is interesting to find that AMP algorithm can converge to an acceptable SO solution using only one iteration. However, the imaging performance obtained by the AMP algorithm is inferior than that of the GPSR algorithm.

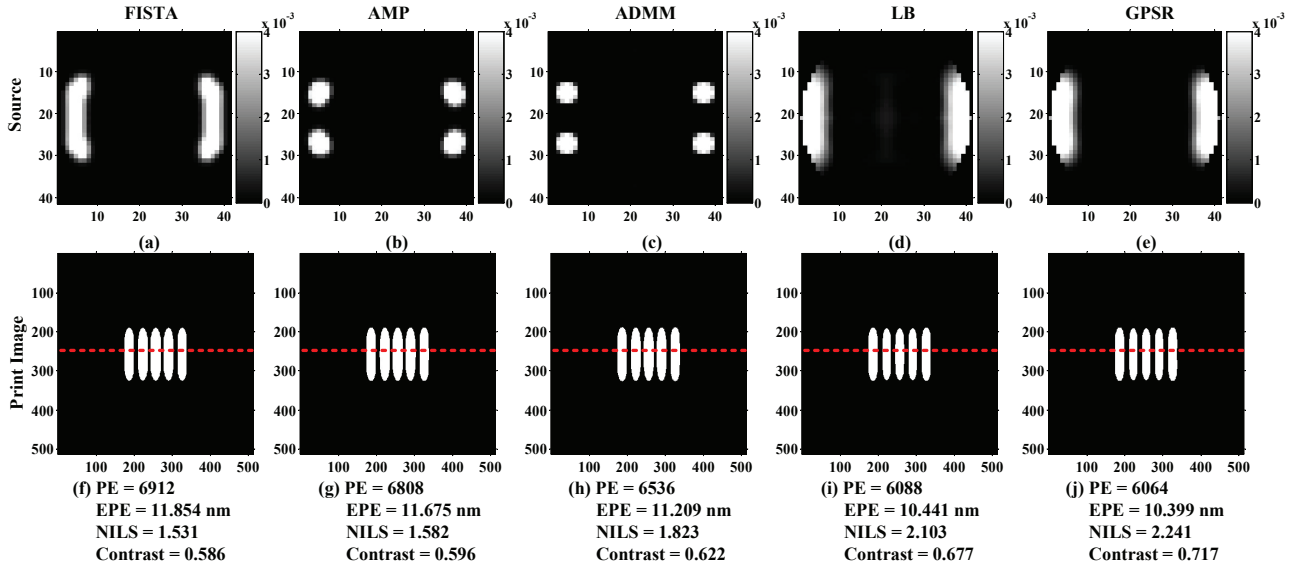

Figure 1. Simulations of different CS algorithms based on the vertical line-space pattern.

In order to further verify the validity of the GPSR algorithm, Fig. 2 illustrates the simulations based on another horizontal block pattern, where the GPSR algorithm can also get better or comparable imaging performance in contrast to other algorithms. Furthermore, the GPSR algorithm is more efficient than the FISTA, ADMM and LB algorithms as shown in Table 1. In addition, Fig. 3 illustrates the simulations of the GPSR algorithm using other two complex patterns. From left to right, it shows the optimized source patterns, mask patterns and the corresponding print images on the focal plane. The top row and bottom row provide the simulations based on different layout patterns. For the layout pattern in Fig. 3(b), the runtime of GPSR algorithm is 0.0237s, while the runtime for the layout pattern in Fig. 3(e) is 0.0748.

Table 1. Comparison of the iteration numbers and runtimes for different CS algorithms.

Vertical Line-Space					
Algorithms	**FISTA**	**AMP**	**ADMM**	**LB**	**GPSR**
Iteration	100	1	283	60	33
Total Time (s)	1.6307	0.0004	0.4823	0.1452	0.0218
Time/Iteration (s)	0.0163	0.0004	0.0018	0.0024	0.0006
Horizontal Block					
Algorithms	**FISTA**	**AMP**	**ADMM**	**LB**	**GPSR**
Iteration	100	1	281	60	31
Total Time (s)	1.7926	0.0036	0.4232	0.1418	0.0597
Time/Iteration (s)	0.0179	0.0036	0.0015	0.0024	0.0019

4.2 Simulations of the CS-SMO algorithm

It is known that the SMO technique can further increase the degree of optimization freedom in contract with the SO or OPC techniques. Combining the proposed CS-SO algorithm with the fast OPC algorithm in Ref. [27], this section develops a fast CS-SMO method, and compares it to the traditional gradient-based SMO method.

Figure 4 illustrates the results of different SMO methods based on the line-space layout pattern. From left to right, it shows the source patterns, mask patterns, and the print images on the focal plane and defocus plane. The first row illustrates the simulation using the initial source pattern and initial mask pattern. The second row shows the result obtained by the conventional gradient-based SMO algorithm proposed in Ref. [28]. The third row illustrates the simulation of the proposed CS-SMO algorithm. In the CS-SMO method, the source pattern is optimized by the GPSR algorithm and the mask pattern is optimized by the iterative hard thresholding algorithm

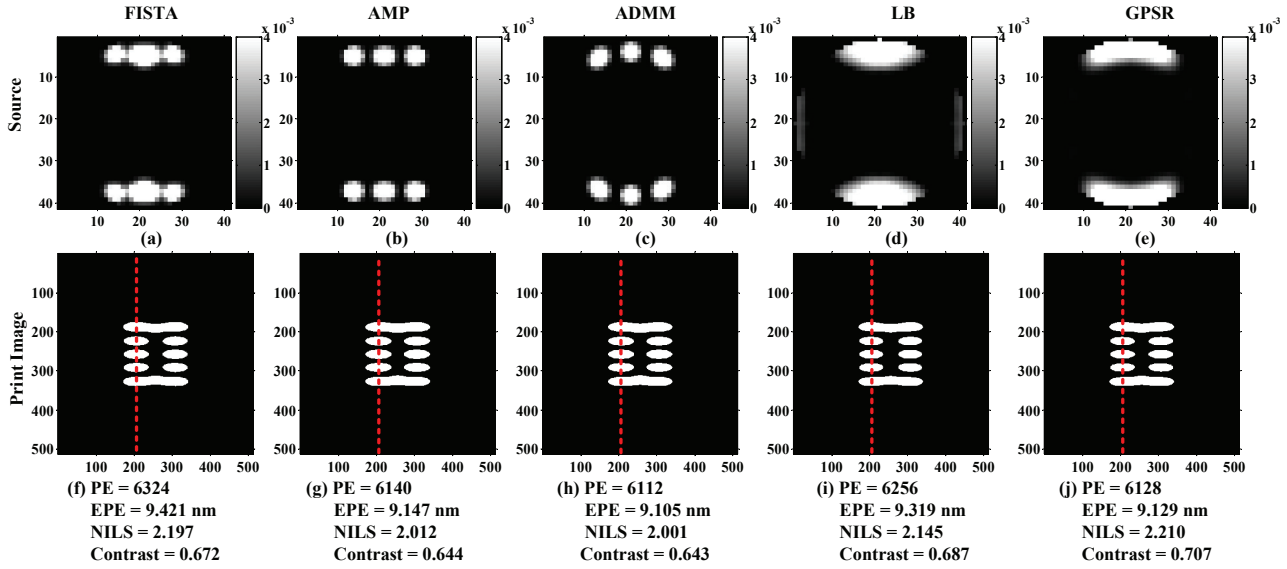

Figure 2. Simulations of different CS methods based on the horizontal block pattern.

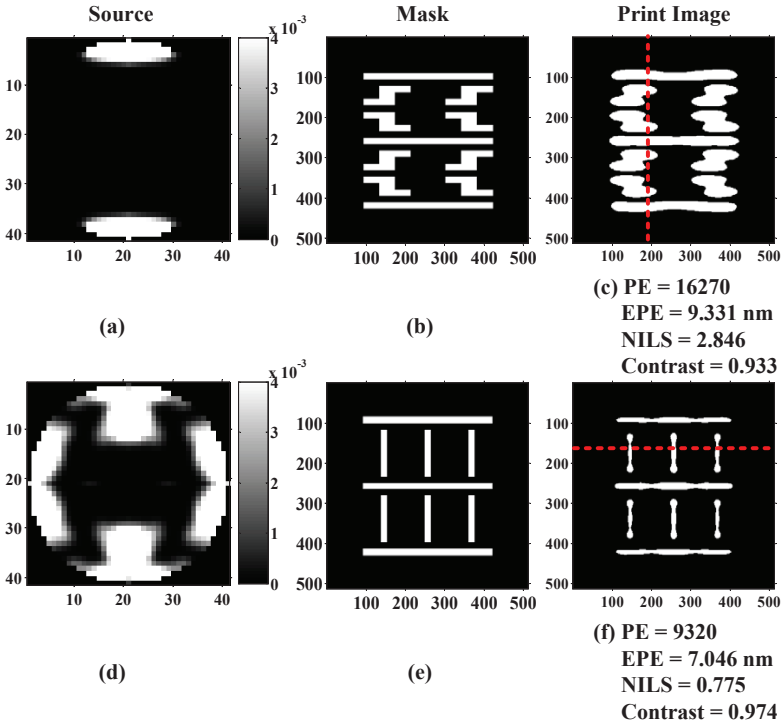

Figure 3. Simulations of the GPSR method based on the complex patterns.

developed in Ref. [27]. During the mask optimization procedure, the data set of layout pattern is downsampled by a factor of 2 ($K = 2$), where K indicates the downsampling rate. The details of the mask optimization algorithm can be found in Ref. [27]. Figure 4 also presents the values of PE, EPE, NILS, and contrast. It is observed that the image fidelity obtained by the proposed CS-SMO method is better than that of the traditional gradient-based SMO method.

Figure 5(a) compares the overlapped process windows (PW) obtained by different SMO methods for the line-space pattern, where the x and y axes represent the depth of focus (DOF) and exposure latitude (EL),

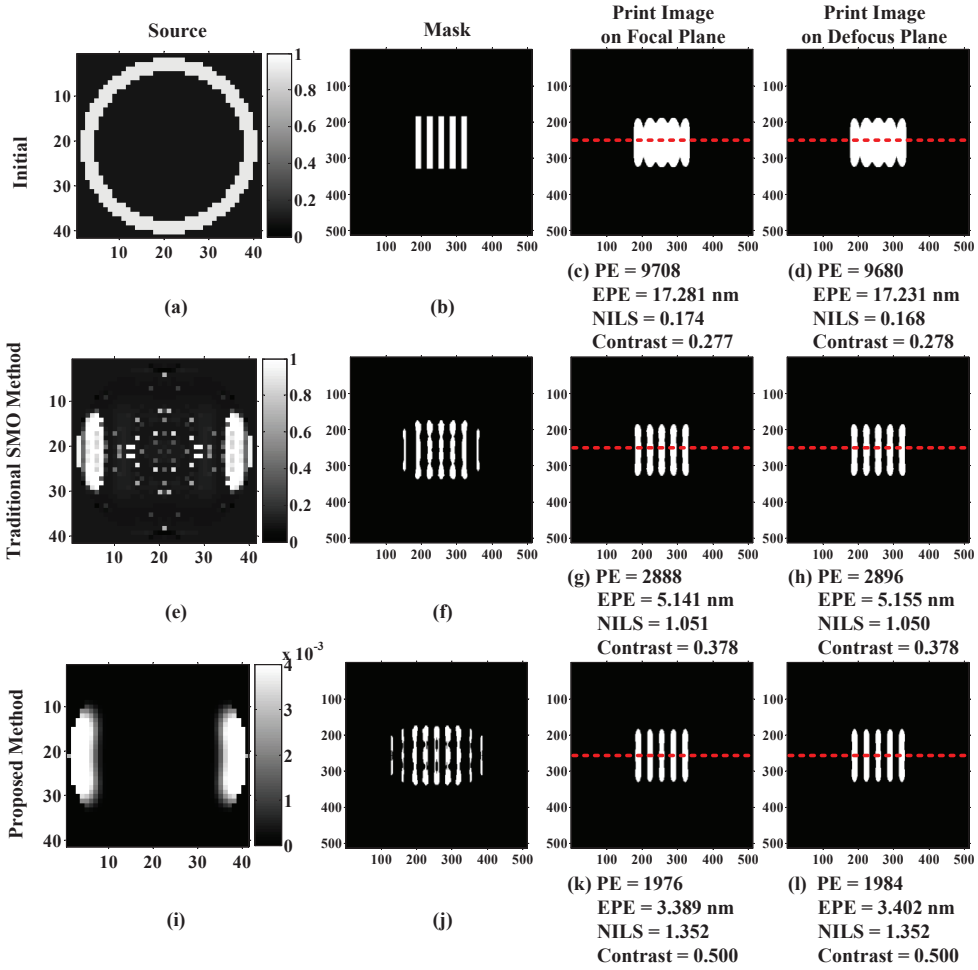

Figure 4. Simulations of the traditional SMO and CS-SMO algorithms based on the vertical line-space pattern.

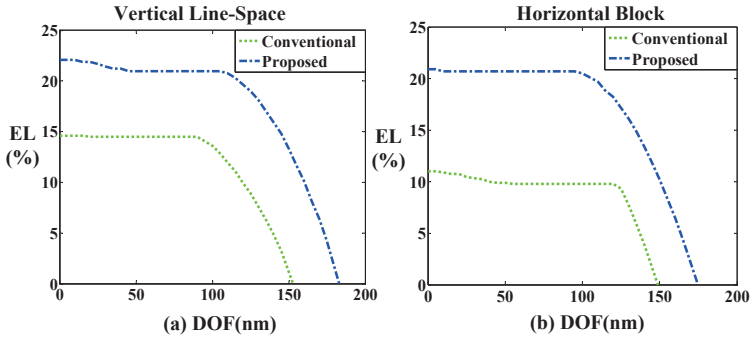

Figure 5. Overlapped PWs obtained by the traditional gradient-based SMO method and the proposed CS-SMO method.

respectively. Figure 6(a) illustrates the measurement positions of the PWs on the layout patterns. It is shown that the proposed CS-SMO method results in larger PW than the traditional gradient-based SMO method.

Next, we compare the computational efficiency of different SMO methods by running the Matlab codes on the same computer. The traditional gradient-based algorithm took 1,170,991s and the proposed CS-SMO method took 73,822s. Thus, the CS-SMO method can improve the computational efficiency by more than one order of magnitude over the gradient-based SMO method.

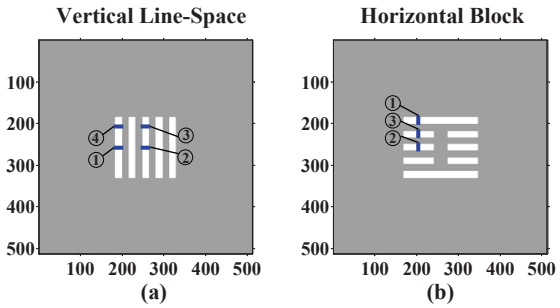

Figure 6. The measurement positions of the PWs.

Figure 7 presents the simulation results of the SMO methods based on another horizontal block pattern. Figure 5 (b) compares the overlapped PWs obtained by both of the SMO methods, and Fig. 6(b) shows the measurement positions of the PWs. Compared to the gradient-based SMO method, the CS-SMO method can effectively improve the image fidelity and process window. For the computational speed, the gradient-based method took 1,261,063s, while the proposed method just took 111,488s.

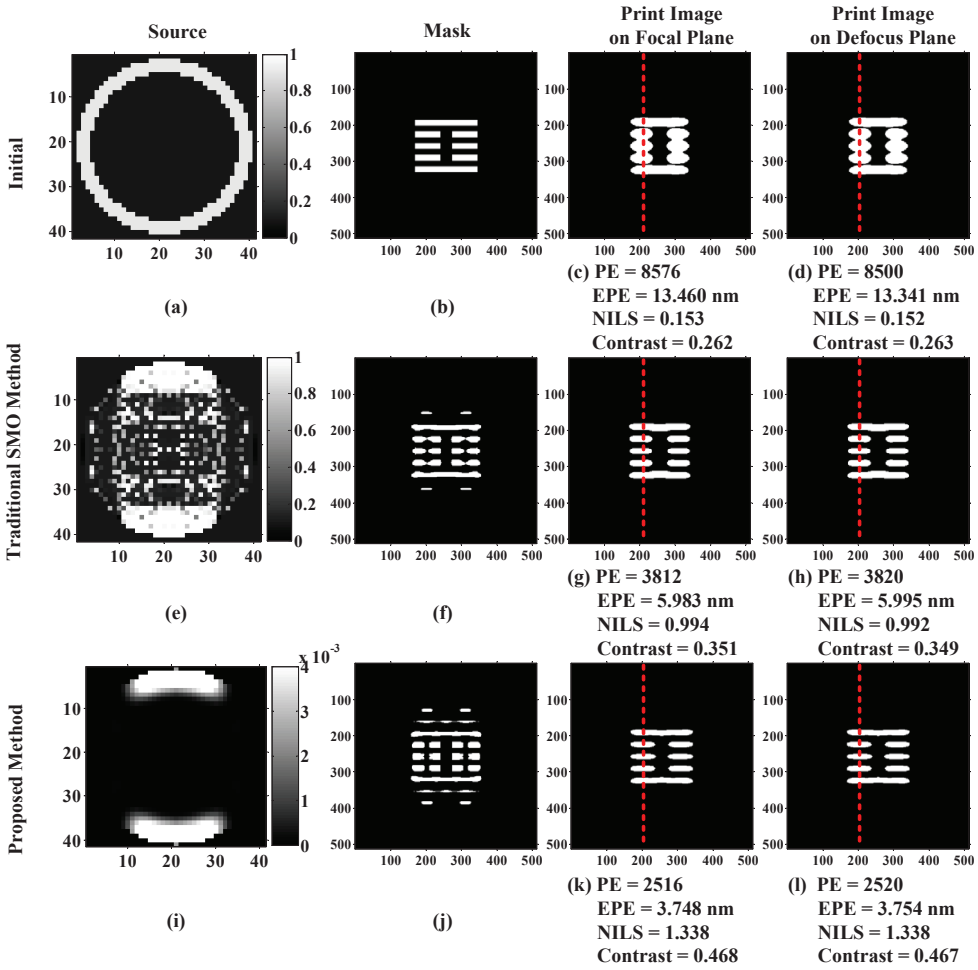

Figure 7. Simulations of the traditional SMO and CS-SMO algorithms based on the horizontal block pattern.

5. CONCLUSION

This paper proposed a fast SO method based on the GPSR algorithm, and compared it with different CS-SO algorithms. Based on the variable splitting and step size searching process, the GPSR method can solve the SO problem efficiently. Meanwhile, the comparison shows that the GPSR method can also achieve superior or comparable imaging performance in contrast to other SO algorithms. In addition, the proposed fast SO algorithm is combined with the OPC method based nonlinear CS to develop an efficient CS-SMO algorithm. The merits of the proposed CS-SMO algorithm compared to the traditional gradient-based algorithm have been demonstrated by a set of simulations.

ACKNOWLEDGMENTS

We thank the financial support by the National Natural Science Foundation of China (NSFC) (61675021), and the Fundamental Research Funds for the Central Universities (2018CX01025).

REFERENCES

1. Wong, A. K., [Resolution Enhancement Techniques in Optical Lithography], SPIE Press, (2001).
2. Ma, X. and Arce, G. R., [Computational Lithography], John Wiley and Sons, New York, 1st ed, (2010).
3. Brist, T. E. and Bailey, G. E., "Effective multicutline QUASAR illumination optimization for SRAM and logic," Proc. SPIE 5042, 153-159 (2003).
4. Burkhardt, M., Yen, A., Progler, C. and Wells, G., "Illuminator design for the printing of regular contact patterns," Microelectronic Engineering s 41-42(3), 91-95 (1998).
5. Gau, T. S., Liu, R. G., Chen, C. K., Lai, C. M. and Liang, F. J., "Customized illumination aperture filter for low k1 photolithography process," Proc. SPIE 4000, 271-282 (2000).
6. Mulder, M., Engelen, A., Noordman, O., Socha, R., Jürgens, D., Trauter, B., Bekaert, J., Laenens, B., Corliss, D. and Mcintyre, G., "Performance of FlexRay: a fully programmable illumination system for generation of freeform sources on high NA immersion systems," Proc. SPIE 7640(3), 511–519 (2010).
7. Granik, Y., "Source optimization for image fidelity and throughput," Journal of Microlithography Microfabrication and Microsystems 3(4), 509-522 (2004).
8. Tian, K., Krasnoperova, A., Melville, D., Rosenbluth, A. E., Gil, D., Tirapu-Azpiroz, J., Lai, K., Bagheri, S., Chen, C. C. and Morgenfeld, B., "Benefits and trade-offs of global source optimization in optical lithography," Proc. SPIE 7274, 72740C (2009).
9. Iwase, K., Bisschop, P. D., Laenens, B., Li, Z., Gronlund, K., Adrichem, P. V. and Hsu, S., "A new source optimization approach for 2x node logic," Proc. SPIE 8166, 81662A (2011).
10. Yu, J. C., Yu, P. and Chao H. Y., "Fast source optimization involving quadratic line-contour objectives for the resist image," Optics Express 20(7), 8161-8174 (2012).
11. Wang, L., Li, S., Wang, X., Yan, G. and Yang, C., "Source optimization using particle swarm optimization algorithm in optical lithography," Optics Express 35(4), 0422002 (2015).
12. Yu, J. and Yu, P., "Gradient-based fast source mask optimization (SMO)," Proc. SPIE 7973(23), 23067–23077 (2011).
13. Li, J., Liu, S. and Lam, E. Y., "Efficient source and mask optimization with augmented lagrangian methods in optical lithography," Optics Express 21(7), 8076-8096 (2013).
14. Ma, X. and Arce, G. R., "Pixel-based simultaneous source and mask optimization for resolution enhancement in optical lithography," Optics Express 17(7), 5783-5793 (2009).
15. Ma, X., Han, C., Gao, J., Li, Y. and Arce, G. R., "Gradient-based joint source polarization mask optimization for optical lithography," Journal of Micro-Nanolithography MEMS and MOEMS 14(2), 023504 (2015).
16. Han, C., Li, Y., Dong, L., Ma, X. and Guo, X., "Inverse pupil wavefront optimization for immersion lithography," Applied Optics 53(29), 6861-6871 (2014).
17. Donoho, D., "Compressive sensing," IEEE Transactions on Information Theory 52(4), 7131-7149 (2006).
18. Candès, E., Romberg, J. and Tao, T., "Robust uncertainty principles: exact signal frequency information," IEEE Transactions on Information Theory 52(2), 489-509 (2006).

19. Ma, X., Wang, Z., Lin, H., Li, Y., Arce, G. R. and Zhang, L., "Optimization of lithography source illumination arrays using diffraction subspaces," Optics Express 26(4), 3738-3755 (2018).
20. Ma, X., Shi, D., Wang, Z., Li, Y. and Arce, G. R., "Lithographic source optimization based on adaptive projection compressive sensing," Optics Express 25(6), 7131-7149 (2017).
21. Osher, S., Burger, M., Goldfarb, D., Xu, J. and Yin, W., "An iterative regularization method for total variation-based image restoration," Multiscale Modeling and Simulation 4(2), 460-489 (2005).
22. Cai, J. F., Osher, S. and Shen, Z., "Linearized bregman iterations for compressed sensing," Mathematics of Computation 78(267), 1515-1536 (2009).
23. Figueiredo, M. A. T., Nowak, R. D. and Wright, S. J., "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," IEEE Journal of Selected Topics in Signal Processing 1(4), 586-597 (2008).
24. Xie, S. and Rahardja, S., "Alternating direction method for balanced image restoration," IEEE Transactions on Image Processing 21(11), 4557-4567 (2012).
25. Beck, A. and Teboulle, M., "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," Siam Journal on Imaging Sciences 2(1), 183-202 (2009).
26. Donoho, D. L., Maleki, A. and Montanari, A., "Message-passing algorithms for compressed sensing," Proceedings of the National Academy of Sciences of the United States of America 106(45), 18914-9 (2009).
27. Ma, X., Wang, Z., Li, Y., Arce, G. R., Dong, L. and Garcia-Frias, J., "Fast optical proximity correction method based on nonlinear compressive sensing," Optics Express 26(11), 14479-14498 (2018).
28. Ma, X., Han, C., Li, Y., Dong, L. and Arce, G. R., "Pixelated source and mask optimization for immersion lithography," Journal of the Optical Society of America A Optics Image Science Vision 30(1), 112-123 (2013).
29. Peng, D., Hu, P., Tolani, V., Dam, T., Tyminski, J. and Slonaker, S., "Toward a consistent and accurate approach to modeling projection optics," Optical Microlithography XXIII, 511-519 (2010).