

Sparse nonlinear inverse imaging for shot count reduction in inverse lithography

Xiaofei Wu,¹ Shiyuan Liu,² Wen Lv,² and Edmund Y. Lam^{1,*}

¹*Imaging Systems Laboratory, Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China*

²*State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, China*

[*elam@eee.hku.hk](mailto:elam@eee.hku.hk)

Abstract: Inverse lithography technique (ILT) is significant to reduce the feature size of ArF optical lithography due to its strong ability to overcome the optical proximity effect. A critical issue for inverse lithography is the complex curvilinear patterns produced, which are very costly to write due to the large number of shots needed with the current variable shape beam (VSB) writers. In this paper, we devise an inverse lithography method to reduce the shot count by incorporating a model-based fracturing (MBF) in the optimization. The MBF is formulated as a sparse nonlinear inverse imaging problem based on representing the mask as a linear combination of shots followed by a threshold function. The problem is approached with a Gauss-Newton algorithm, which is adapted to promote sparsity of the solution, corresponding to the reduction of the shot count. Simulations of inverse lithography are performed on several test cases, and results demonstrate reduced shot count of the resulting mask.

© 2015 Optical Society of America

OCIS codes: (110.5220) Photolithography; (110.1758) Computational imaging; (110.4235) Nanolithography; (100.3190) Inverse problems.

References and links

1. J. Mulken, P. Hinnen, M. Kubis, A. Padiy, and J. Benschop, "Holistic optimization architecture enabling sub-14-nm projection lithography," *J. Micro/Nanolith. MEMS MOEMS* **13**, 1–10 (2014).
2. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography* (SPIE, Washington, 2001).
3. E. Y. Lam and A. K. Wong, "Computation lithography: virtual reality and virtual virtuality," *Opt. Express* **17**, 12259–12268 (2009).
4. E. Y. Lam and A. K. Wong, "Nebulous hotspot and algorithm variability in computation lithography," *J. Micro/Nanolith. MEMS MOEMS* **9**, 033002 (2010).
5. T. Fühner, A. Erdmann, and S. Seifert, "Direct optimization approach for lithographic process conditions," *J. Micro/Nanolith. MEMS MOEMS* **6**, 031006 (2007).
6. B.-G. Kim, S. S. Suh, B.-S. Kim, S.-G. Woo, H.-K. Cho, V. Tolani, G. Dai, D. Irby, K. Wang, G. Xiao, D. Kim, K.-H. Baik, and B. Gleason, "Trade-off between inverse lithography mask complexity and lithographic performance," *Proc. SPIE* **7379**, 73791M (2009).
7. J. Choi, I. K. Shin, and C.-U. Jeon, "Computational technique to overcome the limit of a photomask writer," *J. Micro/Nanolith. MEMS MOEMS* **13**, 023002 (2014).
8. L. Pang, P. Hu, D. Peng, D. Chen, T. Cecil, L. He, G. Xiao, V. Tolani, T. Dam, K.-H. Baik, and B. Gleason, "Source mask optimization (SMO) at full chip scale using inverse lithography technology (ILT) based on level set methods," *Proc. SPIE* **7520**, 75200X (2009).
9. Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," *Opt. Express* **17**, 23690–23701 (2009).

10. A. Poonawala and P. Milanfar, "Mask design for optical microlithography – an inverse imaging problem," *IEEE Trans. Image Process.* **16**, 774–788 (2007).
11. N. Jia, A. K. Wong, and E. Y. Lam, "Regularization of inverse photomask synthesis to enhance manufacturability," *Proc. SPIE* **7520**, 752032 (2009).
12. X. Ma and G. R. Arce, "Binary mask optimization for inverse lithography with partially coherent illumination," *J. Opt. Soc. Am. A* **25**, 2960–2970 (2008).
13. X. Ma and Y. Li, "Resolution enhancement optimization methods in optical lithography with improved manufacturability," *Journal of Micro/Nanolithography, MEMS and MOEMS* **10**, 023009 (2011).
14. X. Ma, Z. Song, Y. Li, and G. R. Arce, "Block-based mask optimization for optical lithography," *Appl. Opt.* **52**, 3351–3363 (2013).
15. W. Lv, Q. Xia, and S. Liu, "Mask-filtering-based inverse lithography," *J. Micro/Nanolith. MEMS MOEMS* **12**, 043003 (2013).
16. X. Wu, S. Liu, W. Lv, and E. Y. Lam, "Robust and efficient inverse mask synthesis with basis function representation," *J. Opt. Soc. Am. A* **31**, B1–B9 (2014).
17. H. Imai and T. Asano, "Efficient algorithms for geometric graph search problems," *SIAM J. Sci. Comput.* **15**, 478–494 (1986).
18. A. B. Kahng, X. Xu, and A. Zelikovsky, "Yield- and cost-driven fracturing for variable shaped-beam mask writing," *Proc. SPIE* **5567**, 360–371 (2004).
19. S. Jiang, X. Ma, and A. Zakhor, "A recursive cost-based approach to fracturing," *Proc. SPIE* **7973**, 79732P (2011).
20. A. Fujimura, I. Bork, T. Kiuchi, T. Komagata, Y. Nakagawa, K. Hagiwara, and D. Hara, "Improvement of mask write time for curvilinear assist features at 22nm," *Proc. SPIE* **7823**, 782307 (2010).
21. B. Yu, J.-R. Gao, and D. Z. Pan, "L-shape based layout fracturing for e-beam lithography," (2013), *Proc. Asia and South Pacific Design Automation Conference*, pp. 249–254.
22. J. Word, Y. Granik, M. Medvedeva, S. Rodin, L. Capodieci, Y. Deng, J. Kye, C. Tabery, K. Yoshimoto, Y. Zou, H. Diab, M. Gheith, M. Habib, and C. Zhu, "Inverse vs. traditional OPC for the 22nm node," *Proc. SPIE* **7274**, 72743A (2009).
23. T. Tao, N. Takahashi, and M. Hamaji, "Novel fracturing algorithm to reduce shot count for curvy shape," *Proc. SPIE* **8880**, 88802J (2013).
24. G. S. Chua, W. L. Wang, B. I. Choi, Y. Zou, C. Tabery, I. Bork, T. Nguyen, and A. Fujimura, "Optimization of mask shot count using MB-MDP and lithography simulation," *Proc. SPIE* **8166**, 816632 (2011).
25. R. C. Pack, K. Standiford, T. Lukanc, G. X. Ning, P. Verma, F. Batarseh, G. S. Chua, A. Fujimura, and L. Pang, "Model-based virtual VSB mask writer verification for efficient mask error checking and optimization prior to MDP," *Proc. SPIE* **9235**, 923509 (2014).
26. S. Jiang and A. Zakhor, "Shot overlap model-based fracturing of curvilinear OPC layouts," (2014), vol. 9235 of *Proc. SPIE*, p. 92351V.
27. T. B. Chan, P. Gupta, K. Han, A. A. Kagalwalla, A. B. Kahng, and E. Sahouria, "Benchmarking of mask fracturing heuristics with known optimum," (2014), *International Conference on Computer-Aided Design*, pp. 246–253.
28. X. Wu, S. Liu, and E. Y. Lam, "Computational techniques to incorporate shot count reduction into inverse lithography," (2015), *Semiconductor Technology International Conference (CSTIC)*, pp. 1–3.
29. X. Wu, S. Liu, J. Li, and E. Y. Lam, "Efficient source mask optimization with Zernike polynomial functions for source representation," *Opt. Express* **22**, 3924–3937 (2014).
30. N. Jia and E. Y. Lam, "Pixelated source mask optimization for process robustness in optical lithography," *Opt. Express* **19**, 19384–19398 (2011).
31. J. Li, S. Liu, and E. Y. Lam, "Efficient source and mask optimization with augmented Lagrangian methods in optical lithography," *Opt. Express* **21**, 8076–8090 (2013).
32. A. Aghasi and J. Romberg, "Sparse shape reconstruction," *SIAM J. Imaging Sci.* **6**, 2075–2108 (2013).
33. S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.* **20**, 33–61 (1998).
34. J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer, 2006), 2nd ed.
35. E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," (2007). <http://www.cs.ubc.ca/labs/scl/spgl1>.

1. Introduction

With the slow arrival of extreme ultraviolet lithography (EUV), optical lithography with ArF source remains to be the most cost effective solution for semiconductor manufacturing. For 22nm node and beyond, the small feature size necessitates holistic optimization of the lithography components [1]. Aggressive resolution enhancement such as inverse lithography technique (ILT) plays an important role for the extension of this solution [2, 3].

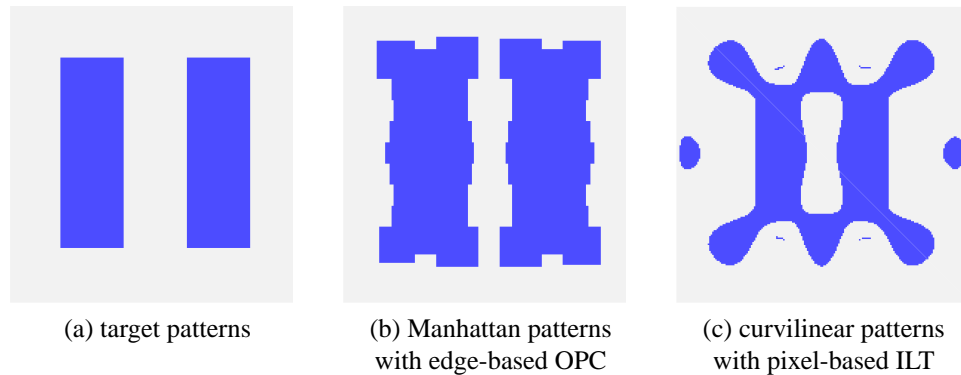


Fig. 1. The mask patterns in optical lithography.

The ILT employs a pixel-based representation of the mask patterns, which has a larger solution space over the conventional edge-based optical proximity correction (OPC) [4,5]. A critical issue of ILT is that the optimized mask patterns can be very complex which are extremely expensive to make. As shown in Fig. 1, the mask patterns created with OPC are usually Manhattan patterns with rectilinear edges, while those resulting from ILT are complex with many curvilinear shapes. The curvilinear patterns can lead to better image quality such as larger process windows, but this is at the expense of a long writing time [6]. Nowadays, the masks are made with writing machines such as the variable shape beam (VSB), which can only write rectangular or triangular shots [7]. To prepare for mask writing, a fracturing process named the mask data preparation (MDP) is performed to partition the mask patterns into rectangles and triangles. The curvilinear patterns in ILT therefore require a large number of shots to write for acceptable fidelity, resulting in a long writing time and high cost. Thus, reducing the shot count for ILT is critical.

Many studies have been done to reduce the mask complexity in ILT. The level set algorithm has been used to solve the ILT and can generate less complex masks, while allowing flexible topographic variations [8, 9]. Regularization methods such as total variation are proposed to reduce the mask complexity and remove the isolated holes [10]. These methods are extended to regularization on mask edges to force them to be rectilinear [11]. The mask manufacturability is also improved with wavelet penalty, topology filter, and block-based algorithms [12–14]. A mask filtering algorithm is developed by Lv *et al.* to remove the mask details and enhance the optimization efficiency as well [15]. The representation methods of mask patterns are changed to discrete cosine function basis functions to reduce the complexity of the optimized patterns [16]. These methods are quite effective to reduce the mask complexity most of the time. However, they do not explicitly incorporate the mask writing process in the optimization, and thus the shot count is not well controlled.

Another approach to reduce the mask making cost is to reduce the shot count during the fracturing process in MDP. The traditional fracturing of Manhattan pattern is formulated as a rectangular recovery problem, and can be solved by an algorithm with the computational cost of $O(n^{1.5\log(n)})$ [17]. In order to take more detailed specification, such as silver reduction, into account, the integer linear programming (ILP) algorithm is introduced to approach it with constraints [18]. A recursive cost-based algorithm is developed by Jiang *et al.* to decrease the external silver length and trapezoid numbers [19]. Other than the fracturing algorithms, the mask writers also evolved from non-overlapping shots to overlapping ones so that the shot count can be reduced [20]. The L-shape shot is recommended to write the mask with a higher

efficiency and fewer shots [21]. However, these algorithms are mostly suitable for Manhattan patterns, and writing curvilinear patterns is still challenging.

To meet this challenge, efforts are made to investigate the mask fracturing techniques along with ILT. It is proposed that the curvilinear patterns can be approximated with Manhattan ones, and then the mask can be manufactured with reasonable cost [22, 23]. However, the shot count is very large, which makes the process expensive. Recently, model-based fracturing (MBF) attracts much attention with its ability to balance the mask fidelity and mask writing cost [20, 24], especially with the need to consider the proximity effect of e-beam writers in the current extremely small scale of optical lithography [25]. An algorithm to fracture the curvilinear patterns is developed, though it still relies on the approximation to edge representation [26]. Another formulation based on integer linear programming method is established by Chan *et al.*, and the problem is solved with the branch and price algorithm [27]. This formulation demonstrates a significant shot count reduction over the traditional methods. Nevertheless, the algorithm is not very efficient and may need hours to fracture a mask pattern even in a very small region. Furthermore, these algorithms consider the mask optimization and fracturing as independent processes, and the MDP does not provide feedback to ILT. To tackle this issue, new process flow is advised to incorporate the mask fracturing to OPC, and computational technique is applied to optimize the mask writer [7, 28]. However, the inefficient algorithm for MBF is computationally challenging to be incorporated into the ILT.

The goal of this paper is to devise an ILT method that incorporates the MBF process for the shot count reduction with an efficient fracturing algorithm. In this method, we represent the mask with a linear combination of a set of basis functions followed by a threshold function to model shot overlapping. The basis functions are defined as rectangular functions corresponding to the shots. The threshold function enforces all the gray values obtained with the linear combination to be 1 or 0, corresponding to the mask intensity. Using this representation, the coefficients can characterize the mask with the basis functions. Therefore, the MBF can be considered a problem to find the appropriate coefficients to recover the mask, and shot count is measured by the number of nonzero coefficients. We formulate this problem as a sparse nonlinear inverse imaging problem aiming to recover the mask pattern and reduce the shot count simultaneously. A Gauss-Newton algorithm is proposed to solve it with a high efficiency, and the algorithm is adapted to promote sparsity of the solution.

In the following, we briefly introduce the ILT problem in Section 2. Then we introduce the mask representation method and the nonlinear inverse imaging formulation of the MBF in Section 3. After that, we perform simulations of ILT incorporating the fracturing process to reduce the mask complexity. Comparisons of the resulting image quality and mask complexity are made over the traditional ILT method to demonstrate its effectiveness.

2. ILT formulation

The main objective of inverse lithography is to design an optimal mask that can produce a target pattern with a forward model. The model transfers the information of a mask pattern to the wafer one as

$$I(x, y) = \mathbf{T}\{M(x, y)\}, \quad (1)$$

where $I(x, y)$ is the resist pattern, (x, y) is the spatial coordinate, $M(x, y) \in \mathbf{R}^{N \times N}$ is the mask pattern, represented by a 2D matrix. The forward model is denoted with $\mathbf{T}\{\cdot\}$, including an imaging process of the optical system and a resist effect, as introduced in previous work [16, 29].

The mask optimization problem is usually considered as an inverse problem, which can be expressed as $\mathbf{T}^{-1}\{\cdot\}$ [10]. This is achieved by solving an optimization problem as

$$M_{opt}(x, y) = \arg \min_{0 \leq M \leq 1} \left\| \mathbf{T}\{M(x, y)\} - I_t(x, y) \right\|_2^2, \quad (2)$$

where $I_t(x,y)$ is the target pattern. The mask intensity is constrained to $[0, 1]$ for binary mask optimization. The process variations such as defocus can be incorporated in the model, as shown in [30, 31].

3. Mask fracturing formulation

3.1. Mask representation

After ILT, the mask undergoes a fracturing process that partitions the mask pattern to some shots to prepare for mask writing. Here we propose a sparse nonlinear inverse imaging formulation for efficient mask fracturing.

We first represent the shot in mask fracturing as a rectangular function

$$S_p(x,y) = \begin{cases} 1, & \text{if } |x-x_p| \leq \frac{W}{2} \text{ and } |y-y_p| \leq \frac{L}{2}, \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where (x_p, y_p) is the position of the center point, and W and L are the width and height of the rectangle. Similar to Chan *et al.*'s algorithm, a library can be formed by the potential shots that can be used to recover the mask pattern [27]. These shots defined by the rectangular functions can include those shots with different sizes and center positions in the region of the target mask.

With the library, the mask is expressed as a linear combination using these rectangular functions as basis functions [29]. After this linear combination, we enforce a threshold function on it, and the mask pattern is represented as

$$M(x,y) = \Gamma \left\{ \sum_{p=1}^K \alpha_p S_p(x,y) - c \right\}, \quad (4)$$

where α_p is the coefficient for a basis function S_p , and K is the total number of basis functions in the library. The coefficient determines whether the potential shot will be written in the mask writing process. If the coefficient is nonzero, it will contribute to the mask recovery, and should be written; otherwise, it will not. The threshold function is denoted by $\Gamma\{\cdot\}$, and c is a value to determine the threshold level. The linear combination value above the threshold c becomes 1 with the threshold function, and it becomes 0 otherwise, corresponding to the intensity of the binary mask. The threshold function can also model the overlapping of two shots. For example, if two shots overlap and both shots produce a gray scale values larger than c , the mask intensities after the threshold are still 1, and will not cause representation errors.

To facilitate numerical computation, the threshold function is approximated with a differentiable one [32]

$$\Gamma(\psi) = \begin{cases} 1, & \psi > \varepsilon \\ 0, & \psi < -\varepsilon, \\ \frac{1}{2} + \frac{\psi}{2\varepsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi\psi}{\varepsilon}\right), & |\psi| \leq \varepsilon \end{cases}, \quad (5)$$

where ψ is a gray scale value, and ε controls the sharpness of the transition. As illustrated in Fig. 2, the region for the transition is determined by ε : a small ε leads to a sharp transition, while a larger value makes the transition a gradual one.

3.2. Sparse nonlinear inverse imaging formulation

With this representation, the mask fracturing becomes the search for those nonzero coefficients α_p that are used to recover the mask pattern. Thus to reduce the shot count, we need to reduce the number of nonzero coefficients, which can be measured by the sparsity of the coefficients.

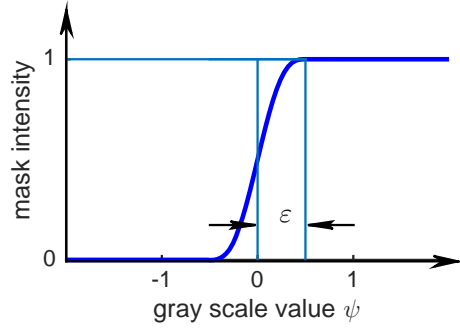


Fig. 2. The threshold function to allow overlapping of the shots.

Therefore, the MBF is formulated as a sparse inverse imaging problem, and the pattern error between the recovered mask and the target one is defined as the cost function

$$\mathbf{E}(\boldsymbol{\alpha}) = \left\| \Gamma \left\{ \sum_{p=1}^K \alpha_p S_p(x, y) - c \right\} - M_t(x, y) \right\|_2^2, \quad (6)$$

where $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_p]^T$ is a vector denoting the coefficients, and M_t is the target pattern to fracture. The ℓ_2 norm indicates the pattern fidelity between the two masks. Then, the problem is formulated as

$$\begin{aligned} & \text{minimize } \mathbf{E}(\boldsymbol{\alpha}), \\ & \text{subject to } \|\boldsymbol{\alpha}\|_0 \leq \sigma, \end{aligned} \quad (7)$$

where the ℓ_0 norm constrained by σ measures the number of nonzero coefficients, corresponding to the shot count.

Equation (7) however is difficult to solve, since the ℓ_0 norm is a hard constraint and non-convex. It is usually relaxed by the ℓ_1 norm, which is convex, and the sparsity can still be achieved [33]. Thus, the problem above is changed to

$$\begin{aligned} & \text{minimize } \mathbf{E}(\boldsymbol{\alpha}), \\ & \text{subject to } \|\boldsymbol{\alpha}\|_1 \leq \tilde{\sigma}. \end{aligned} \quad (8)$$

where $\tilde{\sigma}$ is the maximum ℓ_1 norm to control the sparsity of the coefficients. It can be set to balance the pattern error and the shot count. However, due to the threshold function to model the shot overlapping, it is intrinsically nonlinear. We make use of a Gauss-Newton algorithm to solve this problem and modify the traditional iteration process to promote sparsity.

In the Newton algorithm, the iteration direction δ_k in each iteration is obtained by solving a second order Taylor expansion of the cost function as [34]

$$\delta_k = \arg \min_{\delta} \mathbf{E}(\boldsymbol{\alpha}_k) + \mathbf{J}_{\mathbf{E}}(\boldsymbol{\alpha}_k) \delta + \frac{1}{2} \delta^T \mathbf{H}_{\mathbf{E}}(\boldsymbol{\alpha}_k) \delta, \quad (9)$$

where δ is a variable introduced to calculate the iteration direction, $\mathbf{E}(\boldsymbol{\alpha}_k)$ is the cost function at $\boldsymbol{\alpha}_k$, $\mathbf{J}_{\mathbf{E}}(\boldsymbol{\alpha}_k)$ and $\mathbf{H}_{\mathbf{E}}(\boldsymbol{\alpha}_k)$ are the Jacobian vector and Hessian matrix at $\boldsymbol{\alpha}_k$. We write the cost function as a least square $\mathbf{E}(\boldsymbol{\alpha}) = \|\mathbf{G}(\boldsymbol{\alpha})\|_2^2$, where $\mathbf{G}(\boldsymbol{\alpha})$ is defined as

$$\mathbf{G}(\boldsymbol{\alpha}) = \Gamma \left\{ \sum_{p=1}^K \alpha_p S_p(x, y) - c \right\} - M_t(x, y), \quad (10)$$

the derivatives in Eq. (9) can be written as

$$\mathbf{J}_E \delta = 2 \langle \mathbf{G}'(\boldsymbol{\alpha}) \delta, \mathbf{G}(\boldsymbol{\alpha}) \rangle, \quad (11)$$

$$\frac{1}{2} \delta^\top \mathbf{H}_E(\boldsymbol{\alpha}_k) \delta = \langle \mathbf{G}'(\boldsymbol{\alpha}) \delta, \mathbf{G}'(\boldsymbol{\alpha}) \delta \rangle + \langle \mathbf{G}''(\boldsymbol{\alpha})[\delta, \delta], \mathbf{G}'(\boldsymbol{\alpha}) \rangle, \quad (12)$$

where $\mathbf{G}'(\boldsymbol{\alpha}_k)$, $\mathbf{G}''(\boldsymbol{\alpha}_k)$ are the first and the second order derivatives of $\mathbf{G}(\boldsymbol{\alpha}_k)$ at $\boldsymbol{\alpha}_k$, and $\langle \cdot \rangle$ denotes the inner product. In the Gauss-Newton algorithm, the inner product term containing the second order derivative $\mathbf{G}''(\boldsymbol{\alpha}_k)$ is ignored to reduce the computation cost. Substituting the above expressions in Eq. (9), the quadratic form is equivalent to

$$\|\mathbf{G}(\boldsymbol{\alpha})\|_2^2 + 2 \langle \mathbf{G}'(\boldsymbol{\alpha}) \delta, \mathbf{G}(\boldsymbol{\alpha}) \rangle + \|\mathbf{G}'(\boldsymbol{\alpha}) \delta\|_2^2 = \|\mathbf{G}'(\boldsymbol{\alpha}) \delta + \mathbf{G}(\boldsymbol{\alpha})\|_2^2. \quad (13)$$

Therefore, the iteration direction can be obtained by

$$\delta_k = \arg \min_{\delta} \|\mathbf{G}'(\boldsymbol{\alpha}_k) \delta + \mathbf{G}(\boldsymbol{\alpha}_k)\|_2^2, \quad (14)$$

which is a linear least square problem.

To promote the sparsity of the solution for the optimization problem, we modify the algorithm to search for sparse coefficients [32]. To achieve this, a sparse regularization term is added when solving the above linear problem

$$\begin{aligned} \delta_k &= \arg \min_{\delta} \|\boldsymbol{\alpha}_k + \delta\|_1, \\ \text{subject to } &\|\mathbf{G}'(\boldsymbol{\alpha}_k) \delta + \mathbf{G}(\boldsymbol{\alpha}_k)\|_2^2 \leq \tau, \end{aligned} \quad (15)$$

where the ℓ_1 norm controls the sparsity of $\boldsymbol{\alpha}_k + \delta$, which is the successive coefficient, and τ is a parameter to control the iteration direction. In this way, the coefficients updated by

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \delta_k, \quad (16)$$

are naturally sparse.

The detailed derivation of the first order derivative $\mathbf{G}'(\boldsymbol{\alpha})$ is shown in the Appendix. The optimization expressed in Eq. (15) can be solved by taking advantage of the popular linear sparse basis pursuit algorithms. Thus, the sparsity can be promoted without adding much computation cost from the traditional Gauss-Newton algorithm.

3.3. Illustration of model-based fracturing

To illustrate this idea, we perform the fracturing on a simple mask pattern. As shown in Fig. 3, the mask pattern to fracture in the blue lines is curvilinear. It is represented by a 125×125 pixel image, and the pixel size is 0.5 nm. It should be noted that the number of potential rectangles to recover the mask pattern can be huge even for a very small region, and thus the size of the library formed can be extremely large [27]. To make the algorithm tractable, we limit the shots to be squares with fixed sizes. In this simulation, we set the potential shots as 23×23 nm squares, and the distance between two adjacent rectangles is 1.5 nm, which leads to a shot library with 1600 squares. The maximum value of the ℓ_1 in Eq. (8) is set as 0.5 to limit the number of squares. The value c in the threshold function is 0.5, and ϵ to control the sharpness is set as 0.2. After performing the optimization, the squares selected are merged to form rectangles if they are connected and have parallel edges.

The mask fracturing is conducted with the Gauss-Newton algorithm introduced above, and the linear least square problem formulated in Eq. (15) is solved with a toolbox SPGL1 [35].

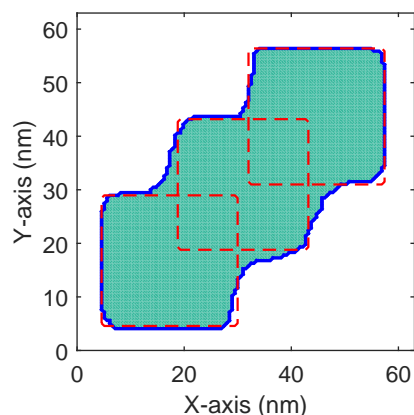


Fig. 3. The mask fracturing result for a typical mask pattern. The blue lines stand for the contour for the mask shape, and the red lines show the shots.

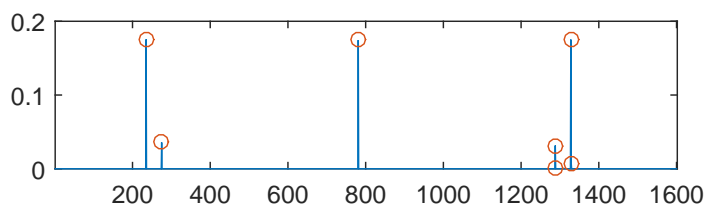


Fig. 4. The values of the representation coefficients, which are shown to be sparse. The optimized result is obtained with 5 iterations within 2.6 seconds.

The fracturing result of the given mask pattern is illustrated in Fig. 3, and the corresponding mask representation coefficients are shown in Fig. 4. The mask shapes are depicted with the blue contours, and the shots are with red rectangles. It is shown that the fracturing result is similar with the benchmark results shown in [27], which is known to be the optimal result. The optimized values of the coefficients are shown in Fig. 4, with most of them are zeros, and those nonzero ones are annotated with red circles. This demonstrates that only several squares are chosen for the representation among the shot candidates, which indicate the proposed sparsity promotion algorithm is effective. The slightly different number of coefficients with the shot count comes from the merging after conducting the optimization.

4. Simulation results

In the following simulations, we focus on incorporating MBF into ILT so that the optimized mask patterns can be less complex. The flow to conduct the inverse lithography is described in Fig. 5. For a given target pattern, we first perform ILT with pixel-based optimization using the conjugate gradient algorithm introduced previously [30]. To provide feedback for mask optimization, we perform a mask fracturing with the Gauss-Newton algorithm introduced above. After this process, another ILT is conducted to generate an optimized pattern with the desired image quality.

In the simulations, the imaging system is set as an immersion one with a quasar illumination source whose inner and outer radii are 0.68 and 0.92, respectively, and the opening angle is 45° . The wavelength of the source is set as 193 nm, and the numerical aperture (NA) of the imaging

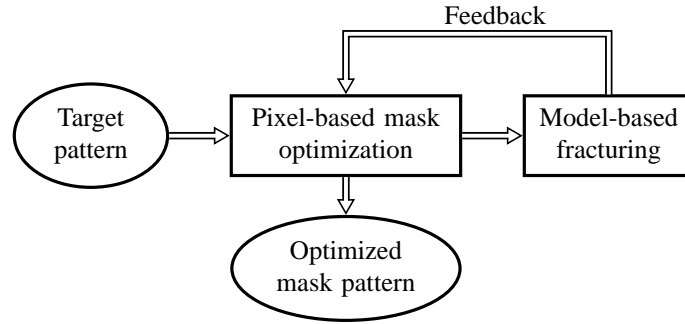


Fig. 5. The flow to incorporate MBF into pixel-based ILT.

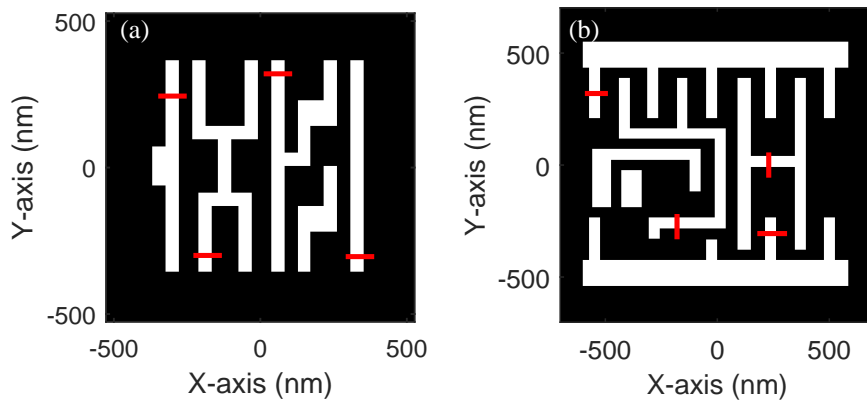


Fig. 6. The target mask patterns for simulations of inverse lithography.

system is set as 1.35. The test patterns are shown in Fig. 6, whose critical dimension (CD) is 45 nm, where the mask on the left is a simple one, and the right one is more complex. They are represented by 301×301 and 401×401 matrices, respectively. Each pixel is 3.5 nm for both masks. The red lines in the figure mark the locations to evaluate the process windows to measure the imaging performance. Similar with our previous work, the process windows are measured as exposure-dose (E-D) window represented by two curves: an upper one corresponding to the doses when the printed CD is 10% smaller than the target one, and a lower curve is 10% larger than it.

In the mask fracturing, we resize the mask patterns to 101×101 pixel image to perform the optimization to avoid the large computation and storage requirement caused by the large library. Similar to the previous illustration, we select the potential shots as 3×3 squares to reconstruct the mask patterns, and the distance between the basis functions equals to 1 pixel. Thus, the total number of basis functions K equals to 9801 in the shot library. Due to the magnification, the actual basis functions to recover the mask are approximately magnified by 3 and 4. The threshold value c and ϵ to control the sharpness is set the same as the example shown in Section 3. The maximum values of the ℓ_1 norm $\tilde{\sigma}$ are 50 and 90, respectively.

We perform simulations following the flow given in Fig. 5 for the first target pattern under the best focus plane. The MBF and ILT in the flow can be iterated for multiple times to improve the image quality and the mask complexity. However, it is at the cost of a larger computation cost. In

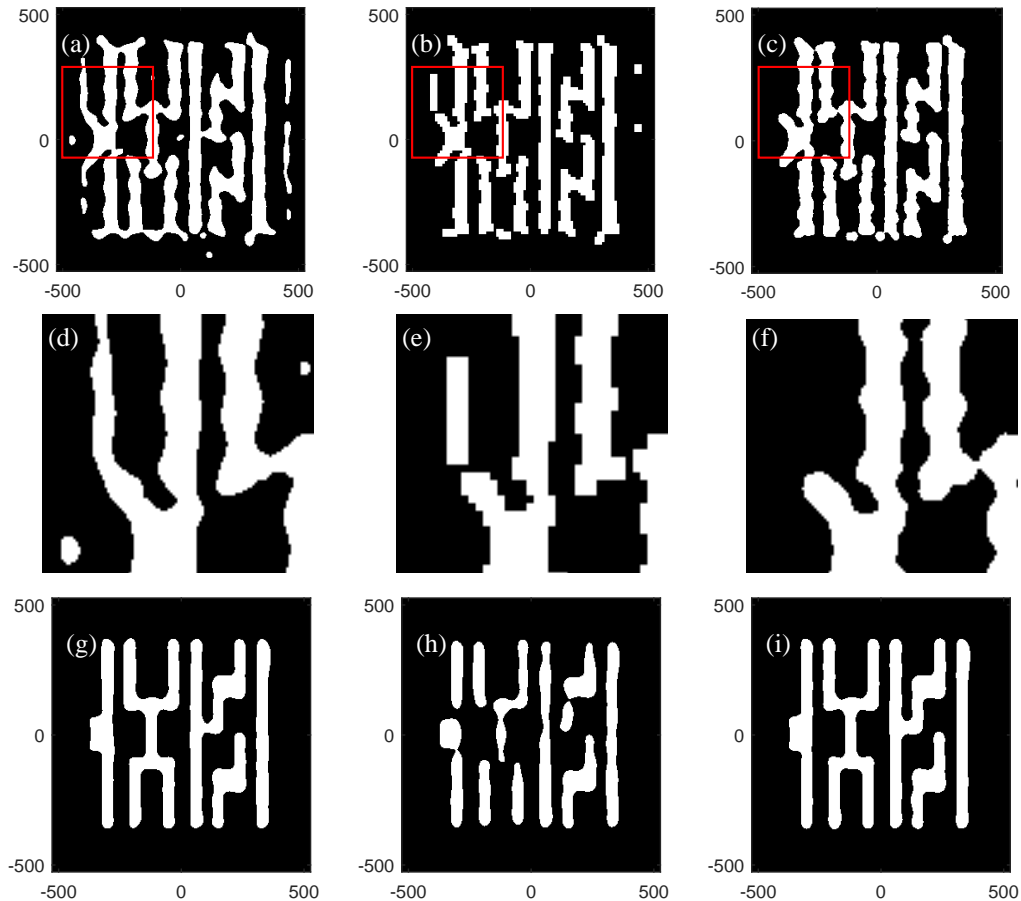


Fig. 7. The mask patterns produced with an initial ILT, a mask fracturing, and the final optimized one are shown in Figs. 7(a)–(c). The magnified views of the patterns annotated by the red rectangles are shown in Figs. 7(d)–(f). The bottom row shows the produced resist patterns corresponding to the mask patterns in the top row.

our simulations, we observe obvious mask complexity reduction if the MBF is performed after the ILT for the first iteration. Even though more times of MBFs are conducted, the reduction of mask complexity is not obvious, and the computation amount increases significantly. Thus, we show the results with only one MBF conducted in the simulations.

For the first test pattern, the produced patterns with initial pixel-based ILT, the one generated with mask fracturing, and final optimized one, are shown in Figs. 7(a)–7(c). Magnified windows marked as red rectangles in these patterns are shown in the lower row as Figs. 7(d)–7(f). The output patterns corresponding to these masks are depicted as Figs. 7(g)–7(i) in the bottom row. The average edge placement error (EPE) of the produced patterns on the wafer delivered by the mask Figs. 7(a) and 7(c) are 2.55 nm and 2.40 nm, indicating similar image quality. The MBF process takes only several seconds, which means the fracturing can be conducted very efficiently. However, another optimization is required after the fracturing, which can make the proposed ILT more computationally intensive than the traditional method.

The pattern shown in Fig. 7(b), which is the recovered one by MBF, is a rectilinear approximation of the optimization result shown in Fig. 7(a). Most of the features in it are rectilinear,

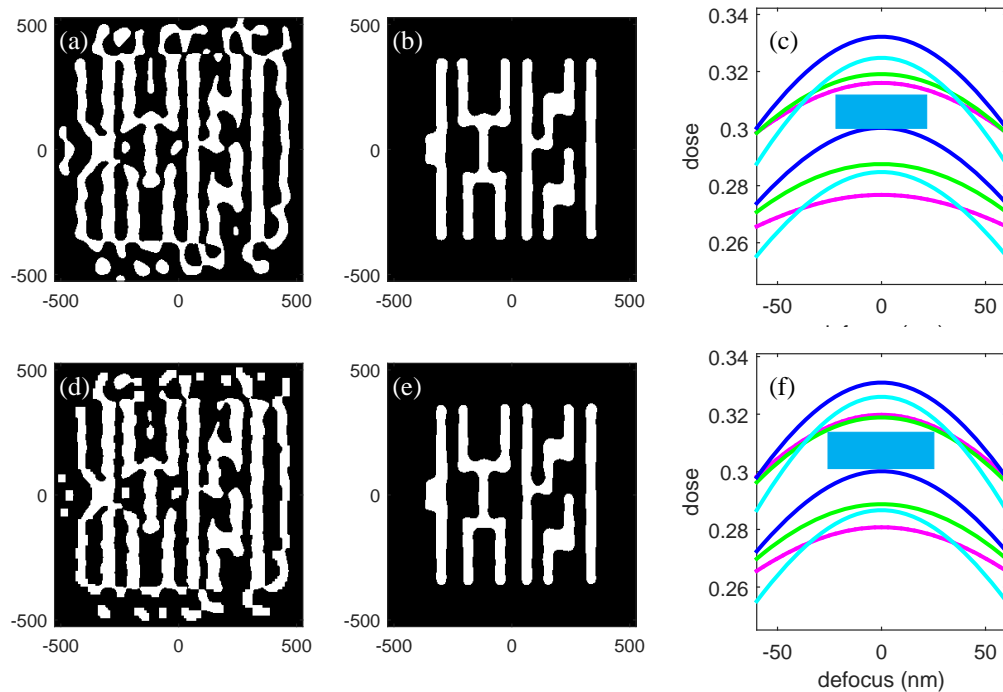


Fig. 8. Simulation results for the first target pattern. Top row shows the optimized mask pattern Fig. 8(a), the produced resist pattern Fig. 8(b), and the measured process windows Fig. 8(c) for the traditional method, while the lower row shows those with the proposed method.

and it captures the important assistant features generated in the ILT. However, it fails to print some of the main features of the target pattern, as shown in Fig. 7(h). The average EPE is not available in several locations since the main features in these places are not printed out. Thus, we conduct another ILT with this pattern as the initial value, and produce an optimized pattern as Fig. 7(c). The produced one becomes more complex compared with Fig. 7(b), but it is less complex than Fig. 7(a). It preserves the rectilinear features such as the rectangles shown in Fig. 7(e), and the line edges are smoother than Fig. 7(a). The similar average EPE it delivered compared with Fig. 7(a) indicates that it has competitive image performance.

Further simulations of ILT are performed for both target patterns under various defocus planes. The range of the defocus is set as 0 to 60 nm with an interval of 10 nm. The optimized patterns for the first target pattern are shown in Figs. 8(a) and 8(d), and 8(b), 8(c), 8(e) and 8(f) show the produced resist patterns and process windows of the two methods, respectively. The process windows are measured as the overlapping area of the exposure-dose (E-D) windows for the measurement places marked as red lines in Fig. 6(a). It is demonstrated in Figs. 8(b) and 8(d) that in both cases the printed patterns at the best focus plane have a good fidelity to the target pattern. The sizes of the overlapping process windows are similar, and the depth of focus (DOF), which is the largest range of the defocus to print an acceptable pattern, are 42 nm and 46 nm, respectively. We make comparison between the optimized patterns shown in Figs. 8(a) and 8(c), and again we observe that the pattern produced with a fracturing method is less complex. It has more rectilinear angles, and some of the assistant features generated are almost rectangles, which can be written with only one shot. However, most of the features shown in Fig. 8(a) are curvilinear, which are expensive to write.

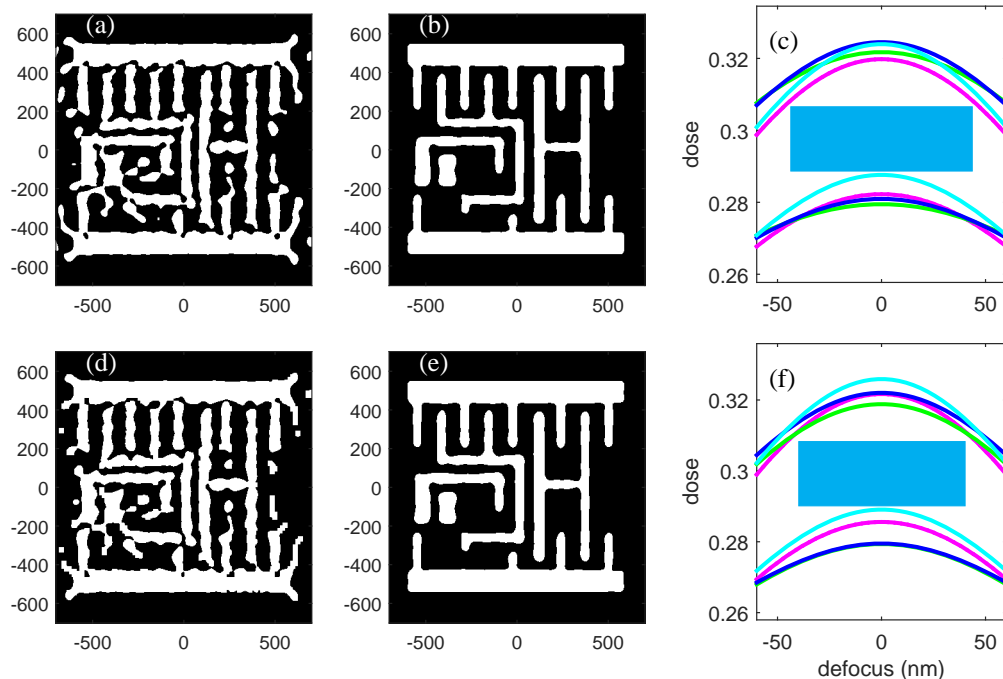


Fig. 9. Simulation results for the second test pattern. Optimization results such as optimization pattern Fig. 9(a), resist pattern Fig. 9(b) and process windows Fig. 9(c) for the traditional ILT are shown in the top row, and those for the proposed ILT method are shown in the lower row.

Similar simulations are conducted for the second target pattern. The optimized patterns with and without the fracturing are shown in Figs. 9(a) and 9(d), and the corresponding resist patterns at the best focus plane and the overlapping process windows are shown in Figs. 9(b), 9(c), 9(e), and 9(f), respectively. It is again observed that the optimized pattern shown in Fig. 9(c) is simpler than the one in Fig. 9(a) with more rectilinear features, and less isolated points. Both patterns can print the main features of the target pattern with good fidelity. The sizes of the overlapping process windows measured at the four locations shown in Fig. 6(b) are similar, and the depth of focus are 84nm and 82nm. The simulation results again demonstrate the MBF proposed can be helpful to reduce the shot count of the optimized patterns while preserving the image quality.

5. Conclusions

This paper proposes to incorporate an MBF process in ILT to reduce the shot count of the optimized mask patterns. The MBF process is formulated as a sparse nonlinear inverse imaging problem, which aims to produce a pattern that is faithful to the target one and acceptable for mask writing simultaneously. The problem is then solved efficiently by a Gauss-Newton algorithm that can promote sparsity in the iterations. Simulations performed show the proposed algorithm can obtain similar fracturing result compared with known optimum, and incorporating this process into ILT is effective to reduce the shot count while producing competitive image performance.

A. Appendix: Gradients derivation

In the following we explain how to compute the derivative $\mathbf{G}(\boldsymbol{\alpha}_k)$ needed to solve the linear sparse basis pursuit problem in Eq. (14), dropping index k for brevity. To facilitate the computation, we rewrite the expression of $\mathbf{G}(\boldsymbol{\alpha})$ in Eq. (10) as

$$\mathbf{G}(\boldsymbol{\alpha}) = \Gamma(\mathbf{S}\boldsymbol{\alpha} - c) - M_t(x, y), \quad (17)$$

where \mathbf{S} is a $N^2 \times K$ matrix generated by stacking the vector form of basis rectangular function $S_p(x, y)$ together. Then the derivative can be computed as

$$\mathbf{G}'(\boldsymbol{\alpha}) = \text{diag} \left[\Gamma'(\mathbf{S}\boldsymbol{\alpha} - c) \right] \mathbf{S}, \quad (18)$$

where $\text{diag}[\cdot]$ transform a vector of the size $N^2 \times 1$ to a diagonal matrix of the size $N^2 \times N^2$. The values in the diagonal of the matrix is filled with the vector, and others are zeros. The derivative the threshold function in Eq. (5) is

$$\Gamma'(\psi) = \begin{cases} 0, & |\psi| > \varepsilon \\ \frac{1}{2\varepsilon} + \frac{1}{2\varepsilon} \cos\left(\frac{\pi\psi}{\varepsilon}\right), & |\psi| \leq \varepsilon \end{cases}. \quad (19)$$

Acknowledgments

This work was supported in part by the UGC Areas of Excellence project Theory, Modeling, and Simulation of Emerging Electronics, and by the State Key Lab of Digital Manufacturing Equipment and Technology under Project DMETKF2013003. We also would like to acknowledge funding from the National Natural Science Foundation of China (Grant Nos. 51475191 and 51405172) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120142110019).