Resolution enhancement optimization methods in optical lithography with improved manufacturability

Xu Ma
Yanqiu Li
Resolution enhancement optimization methods in optical lithography with improved manufacturability

Xu Ma
Yanqiu Li
Beijing Institute of Technology
Key Laboratory of Photoelectronic Imaging
Technology and System of Ministry of Education of China
School of Optoelectronics
Beijing, 100081 China
E-mail: maxu@bit.edu.cn

Abstract. Optical proximity correction (OPC) methods are resolution enhancement techniques used extensively in the semiconductor industry to improve the resolution and pattern fidelity of optical lithography. During the mask data preparation process, the mask pattern is first fractured into basic rectangles, and then fabricated by the variable-shaped-beam mask writing machine. The rectangle count included in the fractured pattern is preferable to be suppressed to reduce the mask fabricating time and cost. Recently, various pixel-based OPC (PBOPC) approaches have been developed to improve the resolution of optical lithography systems. However, these approaches fall short in controlling the rectangle count in the fractured pattern, thus deteriorating the manufacturability of the mask. This paper focuses on developing gradient-based PBOPC optimization algorithms to improve the resolution of optical lithography, while controlling the manufacturability of the mask. To achieve this goal, a topography filter is designed to analytically formulate the rectangle count in the fractured pattern during the optimization process. The manufacturability cost term is then introduced to constrain the complexity of the mask. Cost sensitivity is applied to speed up the proposed algorithms. A line search method is used to properly choose the parameters, and leads to superior resolution and manufacturability of masks. © 2011 Society of Photo-Optical Instrumentation Engineers (SPIE). [DOI: 10.1117/1.3590252]

Subject terms: optical lithography; resolution enhancement; optical proximity correction; manufacturability; fracture; partially coherent imaging system.

Paper 11002RR received Jan. 1, 2011; revised manuscript received Apr. 19, 2011; accepted for publication Apr. 20, 2011; published online May 31, 2011.

1 Introduction

Due to the resolution limits of optical lithographic systems, the electronics industry has relied on resolution enhancement techniques (RETs) to compensate and minimize imaging distortions as the mask patterns are projected onto semiconductor wafers.1,2 Resolution in optical lithography obeys the Rayleigh resolution limit \( R = \frac{k \lambda}{NA} \), where \( \lambda \) is the wavelength, NA is the numerical aperture, and \( k \) is the process constant which can be minimized through RET methods.3–6

In optical proximity correction (OPC), mask amplitude patterns or adding the sub-resolution assist features that can pre-compensate for imaging distortions.1,2

In general, the OPC approaches are divided into two classes: rule-based and model-based approaches.2 Rule-based approaches are mostly heuristic, and simple to implement, but can only compensate the warping in local features. On the other hand, model-based approaches use mathematical models to represent the image formation process of the optical lithography system, and iteratively seek the global minimization of the cost function to improve the image fidelity on the wafer. There are two types of model-based OPCs: edge-based OPC (EBOPC) and pixel-based OPC (PBOPC). An EBOPC decomposes the mask into edges and corners and optimizes their locations, while a PBOPC decomposes the mask into small pixels and optimizes each pixel.2 Recent studies have shown that the EBOPC may not be very suitable for 45nm and smaller technology nodes due to the inability to generate assist features, the less degree of freedom during the optimization process and so on.7 Compared to EBOPC, PBOPC has more degrees of freedom during the optimization process, and may result in higher resolution of the projected image on the wafer. Thus, PBOPC approaches gain a revival of interest for the advanced lithography with technology nodes under 45 nm.

However, the aggressive use of PBOPC poses new challenges to the manufacturability of the mask pattern.8 During the mask data preparation process, the optimized masks obtained from OPC approaches are initially fractured into numerous rectangles. Subsequently, these rectangles are exposed by the variable-shaped-beam (VSB) mask writing machine.9 First, the turned off electron-beam moves on a vector path and directly reaches the rectangle to be exposed. Then, the electron-beam is turned on to expose the nether rectangle on the PBOPC. After that, the electron-beam is turned off and moves again until the entire pattern is completely exposed. Smaller rectangles in the fractured mask pattern are exposed by one shot of the electron-beam, while larger ones are exposed by multiple shots. Thus, the rectangle count included in the fractured mask pattern is preferable to be suppressed in order to reduce the mask fabricating time and cost. Since each pixel on the PBOPC may be flipped independently, the masks obtained from the PBOPC approaches are usually geometrically complex. Therefore, the PBOPC approaches dramatically increase the rectangle count in the fractured mask pattern and deteriorate the manufacturability of the mask pattern. An example is shown in Fig. 1.
Figure 1(a) is the mask pattern without PBOPC for 4 line-shapes, which includes 4 rectangles. Figure 1(b) is the corresponding PBOPC pattern. The PBOPC introduces four serifs at the line ends and two assist bars besides the main bodies. These assist features are used to correct the rounding effect of line ends and the shrinking of line width, respectively. However, the PBOPC increases the rectangle count from 4 to 14, where the dashed lines show the fracture of the entire PBOPC pattern. Hence, one critical issue in PBOPC is the manufacturability of the optimized masks.

In the past, several PBOPC optimization algorithms have been proposed in the literature. Sherif et al. derived an iterative approach to generate binary masks in incoherent diffraction-limited imaging systems. Liu and Zakhor developed a binary and phase shifting mask design strategy based on the branch and bound algorithm and simulated annealing. Erdmann et al. proposed the optimization of the mask and illumination parameters with a genetic algorithm. Granik described and compared solutions of inverse mask problems. Rosenbluth et al. first proposed the simultaneous source and mask optimization algorithm. However, the searching process of the methods mentioned above for a suitable solution is either computationally expensive or not efficient. Poonawala and Milanfar recently introduced a line search method of the parameters are illustrated in Sec 7. Conclusions are provided in Sec 8.

2 Fourier Series Expansion Model of Partially Coherent Imaging Systems

According to the Hopkins imaging model, the light intensity distribution exposed on the wafer in partially coherent imaging systems is bilinear and described by

$$I(r) = \int \int M(r_1)M(r_2)\gamma(r_1 - r_2)h^*(r - r_1) \times h(r - r_2)dr_1dr_2,$$

where $r = (x, y)$ is the coordinate on the image plane located on the wafer, $r_1 = (x_1, y_1)$ and $r_2 = (x_2, y_2)$ are the coordinates on the object plane located on the mask. $M(r)$ is the mask pattern, $\gamma(r_1 - r_2)$ is the complex degree of coherence, and $h(r)$ represents the amplitude impulse response of the optical system. The complex degree of coherence $\gamma(r_1 - r_2)$ is generally a complex number, whose magnitude represents the extent of optical interaction between two spatial locations $r_1 = (x_1, y_1)$ and $r_2 = (x_2, y_2)$ of the light source. The complex degree of coherence in the spatial domain is the inverse 2D Fourier transform of the illumination shape. A schematic of an optical lithography system with partially coherent illumination is illustrated in Fig. 2. The light source with a wavelength of $\lambda$ is placed at the focal plane of the first condenser, illuminating the mask. Common illumination sources include dipole, quadrupole, and annular shapes, all introducing partial coherence. The image of the photomask is formed by the projection optics onto the wafer.
coherence factor $\gamma = a/b$ is defined as the ratio between the size of the source image and the pupil.

In order to formulate the PBOPC optimization problem with partially coherent illumination, the Fourier series expansion model is discussed in the following.\(^{26}\) Assume the mask is constrained in the square area $A_m$ defined by $x, y \in [-D/2, D/2]$. Thus, for the computations involved in Eq. (1), the only values of $\gamma(r)$ needed are those inside the square area $A_m$. Applying the 2D Fourier series expansion, $\gamma(r)$ can be rewritten as

$$\gamma(r) = \sum_m \Gamma_m \exp(j \omega_0 m \cdot r), \quad (2)$$

and

$$\Gamma_m = \frac{1}{D^2} \int_{A_m} \gamma(r) \exp(j \omega_0 m \cdot r) dr, \quad (3)$$

where $\omega_0 = \pi/D$, $m = (m_x, m_y)$, $m_x$ and $m_y$ are integers, and $\cdot$ is the inner-product operation. Substituting Eq. (2) into Eq. (1), the light intensity on the wafer is given by

$$I(r) = \sum_m \Gamma_m |M(r) \otimes h_m(r)|^2, \quad (4)$$

where

$$h_m(r) = h(r) \exp(j \omega_0 m \cdot r). \quad (5)$$

It is observed from Eq. (4) that the PCI is equal to the superposition of coherent systems. Since the Fourier series expansion model is based on direct discretization of the Hopkins imaging model, they have the same accuracy.

For the annular illumination, the complex degree of coherence is

$$\gamma(r) = \frac{J_1(2\pi r/2D_{cu})}{2\pi r/2D_{cu}} \frac{J_1(2\pi r/2D_{el})}{2\pi r/2D_{el}}, \quad (6)$$

where $r = \sqrt{x^2 + y^2}$. The corresponding Fourier series coefficients are

$$\Gamma_m = \begin{cases} \frac{4J_1^2(D_{el}^2 - D_{cu}^2)}{\pi^2(D_{el}^2 - D_{cu}^2)} & \text{for } D/2D_{el} \leq |m| \leq D/2D_{cu}, \\ 0 & \text{elsewhere} \end{cases} \quad (7)$$

where $D_{el}$ and $D_{cu}$ are the coherent lengths of the inner and outer circles respectively, $\sigma_{inner} = \frac{D_{el}}{2NA}$ and $\sigma_{outer} = \frac{D_{cu}}{2NA}$ are the corresponding inner and outer partial coherence factors. The convolution kernel $h(r)$ is defined as the Fourier transform of the circular lens aperture with cutoff frequency $NA/\lambda$.\(^{27,28}\) Therefore,

$$h(r) = \frac{J_1(2\pi r NA/\lambda)}{2\pi r NA/\lambda}. \quad (8)$$

### 3 Lithography Preliminaries

Let $M(x, y) = 0$ or 1 be the input binary mask. $T\{\cdot, \cdot\}$ denotes an optical lithography system, with a partially coherent illumination. The PCI optical system is approximated by a Fourier series expansion model shown in Eq. (4). The effect of the photoresist is modeled by a hard threshold operation. The output pattern is denoted as $Z(x, y) = T\{M(x, y)\}$. Given a $N \times N$ desired output pattern $\tilde{Z}(x, y)$, the goal of the PBOPC optimization is to find the optimized $M(x, y)$ called $\tilde{M}(x, y)$ such that the distance

$$D = d[Z(x, y), \tilde{Z}(x, y)] = d[T\{M(x, y)\}, \tilde{Z}(x, y)] \quad (9)$$

is minimized, where $d(\cdot, \cdot)$ is the square of the l-2 norm. The mask optimization problem can thus be formulated as

$$\tilde{M}(x, y) = \arg \min_{M(x, y)} d[T\{M(x, y)\}, \tilde{Z}(x, y)]. \quad (10)$$

Figure 3 depicts the approximated forward process model of optical lithography systems, where the Fourier series expansion model is used in the image formation stage, and a hard threshold function is used to approximate the photoresist effect. The output pattern of the optical system is binary. Further, since the derivative of the sigmoid function exists, it is used to approximate the hard threshold function. The sigmoid function is

$$\text{sig}(x) = \frac{1}{1 + \exp[-a(x - t_i)]}, \quad (11)$$

where $t_i$ is the process threshold, and $a$ dictates the steepness of the sigmoid function. Following the definitions above, the following notations are used:

1. The $M_{N \times N}$ matrix represents the mask pattern, with entry values equal to 0 or 1. The $N^2 \times 1$ equivalent raster scanned vector representation is denoted as $m$.

$$m \rightarrow \sum_m |m| = \rightarrow \text{Sig}(\sum_m |m|)$$

![Fig. 3 Approximated forward process model.](image-url)
2. \( h^m \) is the amplitude impulse response of the \( m \)th term in Eqn. (4).
3. The desired \( N \times N \) binary output pattern is denoted as \( Z \). It is the desired intensity distribution sought on the wafer. Its vector representation is denoted as \( \tilde{z} \).
4. The output of the sigmoid function is the \( N \times N \) image denoted as:
   \[
   Z = \text{sig} \left\{ \sum_m \Gamma_m |M \otimes h^m|^2 \right\}.
   \] (12)
   The equivalent vector is denoted as \( \tilde{z} \). The \( i \)th entry in \( \tilde{z} \) can be represented as
   \[
   \tilde{z}_i = \frac{1}{1 + \exp \left( -a \sum_m \Gamma_m |\sum_{j=1}^N h_{ij}^m m^j|^2 + at \right)},
   \] \( i = 1, \ldots, N^2 \), (13)
where \( h_{ij} \) is the \((i, j)\)th entry of the filter.
5. The hard threshold version of \( Z \) is the binary output pattern denoted as \( Z_b \). Its equivalent vector is denoted as \( \tilde{z}_b \), with all entries constrained to 0 or 1.
6. The optimized mask denoted as \( \hat{M} \) minimizes the distance between \( Z \) and \( \tilde{Z} \), i.e.,
   \[
   \hat{M} = \arg \min_M \left\{ \sum_m \Gamma_m |M \otimes h^m|^2, Z \right\},
   \] (14)
where \( \hat{M} \) has entry values equal to 0 or 1.

4 Formulation of the Manufacturability Properties
4.1 Manufacturing Problem
After the mask optimization process, the optimized mask pattern is fabricated in the mask writing step. The mask pattern is initially fractured into numerous rectangles without overlap. Subsequently, these rectangles are exposed by the VSB mask writing machine. Each rectangle corresponds to at least one shot in the VSB mask writing machine. Thus, the rectangle count in the fractured mask pattern should be minimized to reduce the fabricating time and cost.

In this paper, we consider the mask patterns with only horizontal and vertical edges, which can be represented by a set of rectilinear polygons. We use the term polygon to mean rectilinear polygon in the remainder of this paper for convenience. For the fractured pattern of an individual polygon, it has been proven that the theoretical lower bound of the rectangle count is
\[
\#(\text{rectangle}) = #\text{(concave)} + 1.
\] (15)
where \#(rectangle) is the total rectangle count, \#(concave) is the number of concave vertices, and \#(chord) is the number of chords in the fractured pattern. The chord is defined as a horizontal or vertical line, which connects two concave vertices and does not intersect with other polygon boundaries. Figure 4 shows a fractured pattern with one chord \( \overline{AB} \).

In Fig. 4, there are three concave vertices marked by green points and one chord \( \overline{AB} \). According to Eqn. (15), the rectangle count is three, which is consistent with the fractured pattern in Fig. 4. One necessary condition of the existence of chords is that two concave vertices have the same \( x \) or \( y \) coordinate. However, this condition is rarely met in a practical PBOPC pattern, since every pixel on the PBOPC may be flipped during the optimization process. Thus, the fractured pattern of the mask obtained from PBOPC optimization is not probable to include many chords. In order to simplify the formulation, we ignore \#(chord) and modify Eq. (15) as
\[
\#(\text{rectangle}) = #\text{(concave)} + 1.
\] (16)
As mentioned above, Eq. (16) is only correct for an individual polygon case. If the mask includes several polygons, Eq. (16) should be adjusted as
\[
\#(\text{rectangle}) = #\text{(concave)} + #\text{(polygon)}.
\] (17)
where \#(polygon) is the number of polygons in the mask. For a rectilinear polygon, there is always four more convex vertices than concave vertices. Therefore, Eq. (17) can be modified as
\[
\#(\text{rectangle}) = #\text{(concave)} + \frac{#\text{(convex)} - #\text{(concave)}}{4}
\]
\[
= \frac{3}{4} #\text{(concave)} + \frac{1}{4} #\text{(convex),}
\] (18)
where \#(convex) is the number of convex vertices in the mask. An example is shown in Fig. 1(b), where black and white points represent convex and concave vertices, respectively. There are 32 convex vertices and 8 concave vertices, thus 14 rectangles.

4.2 Topography Filter and Manufacturability Cost
In order to reduce the mask fabricating time and cost, we need to decrease the rectangle count in the fractured mask pattern during the optimization process. In the following, the rectangle count is referred to as the manufacturability cost which is denoted as \( S \). According to Eq. (18), the number of convex and concave vertices are needed to analytically formulate the manufacturability cost. In order to calculate the number of convex and concave vertices, we introduce the topography filter which is defined as
\[
g = I_{2 \times 2} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.
\] (19)
where all components of the matrix are 1. The convolution between the mask and the topography filter is

$$G_M = g \odot M. \quad (20)$$

Its vector representation is denoted as $\hat{g}_M$. For each pixel $(x, y)$, the value of $G_M(x, y)$ has six possibilities as shown in Fig 5. Case I is the empty area leading to $G_M(x, y) = 0$; case II is the convex vertex leading to $G_M(x, y) = 1$; case III is the opposite vertical angle leading to $G_M(x, y) = 2$; case IV is the edge leading to $G_M(x, y) = 3$; Case V is the concave vertex leading to $G_M(x, y) = 4$. One problem is that both Case III and IV result in $G_M(x, y) = 2$, which leads to aliasing. In other words, the value of $G_M(x, y)$ is inadequate to distinguish between the opposite vertical angle and the edge. To remove the aliasing, we prohibit the opposite vertical angle when we optimize the mask pattern. The details of the method to achieve this goal are described in Sec 6. As a result, the value of $G_M(x, y)$ one-to-one corresponds to different local topographic cases.

Based on the relationship between $G_M(x, y)$ and the local topography, the number of convex and concave vertices can be formulated as

$$\text{(convex)} = -\frac{1}{6} \sum_{x=1}^{N} \sum_{y=1}^{N} G_M(x, y) \times [G_M(x, y) - 2][G_M(x, y) - 3][G_M(x, y) - 4]$$

$$= -\frac{1}{6} \sum_{k=1}^{M^2} [g_M(x, y)[g_M(x, y) - 2]$$

$$[g_M(x, y) - 3][g_M(x, y) - 4]$$

$$= -\frac{1}{6} \sum_{j=1}^{N^2} \left( \sum_{j=1}^{N^2} g_{kj}m_j \right) \left( \sum_{j=1}^{N^2} g_{kj}m_j - 2 \right)$$

$$\times \left( \sum_{j=1}^{N^2} g_{kj}m_j - 3 \right) \left( \sum_{j=1}^{N^2} g_{kj}m_j - 4 \right) \quad (21)$$

Similarly,

$$\#(\text{concave}) = -\frac{1}{6} \sum_{k=1}^{M^2} \left[ \left( \sum_{j=1}^{N^2} g_{kj}m_j \right) \left( \sum_{j=1}^{N^2} g_{kj}m_j - 1 \right)$$

$$\left( \sum_{j=1}^{N^2} g_{kj}m_j - 2 \right) \left( \sum_{j=1}^{N^2} g_{kj}m_j - 4 \right) \right]. \quad (22)$$

Substitute Eqs. (21) and (22) into Eq. (18), the manufacturability cost can be calculated as

$$S = \#(\text{rectangle})$$

$$= -\frac{1}{12} \sum_{k=1}^{M^2} \left[ \left( \sum_{j=1}^{N^2} g_{kj}m_j \right) \left( \sum_{j=1}^{N^2} g_{kj}m_j - 2 \right)$$

$$\times \left( \sum_{j=1}^{N^2} g_{kj}m_j - 4 \right) \right]. \quad (23)$$

5 Cost Function and Cost Sensitivity

According to Eqs. (9) and (10), the formulation of the PBOPC optimization problem is relied on the selected cost function $D$. It is desired to simultaneously improve the fidelity of the print image on the wafer and reduce the rectangle count in the fractured mask pattern. Thus, we divide the cost function into two parts: the fidelity cost term $F$ and the manufacturability cost term $S$. The fidelity cost term is formulated as the square of the $l$-2 norm of the difference between $\bar{z}$ and $\bar{z}_{c}$. Therefore,

$$F(m) = \| \bar{z} - \bar{z}_c \|_2^2 = \sum_{i=1}^{N^2} (\bar{z}_c - \bar{z}_c)^2, \quad (24)$$

where $\bar{z}_c$ in Eq. (24) is represented in Eq. (13). On the other hand, the manufacturability cost term is represented in Eq. (23). Thus, the overall cost function is formulated as

$$D = F + \gamma \frac{\| F \|_2}{\| S \|_2} S, \quad (25)$$

where $\gamma$ is a user-defined parameter to reveal the weight of the manufacturability cost term, $\| \cdot \|_2$ is the $l$-2 norm, and $\| F \|_2/\| S \|_2$ is a normalization term. In the following, the gradient of the cost function $D$, referred to as the cost sensitivity, is used to navigate the cost function in the descent direction during the optimization process. The cost sensitivity $\nabla D(M)$ is calculated as:

$$\nabla D(M) = \nabla F(M) + \gamma \frac{\| F \|_2}{\| S \|_2} \nabla S(M), \quad (26)$$

where $\nabla D(M), \nabla F(M), \nabla S(M) \in \mathbb{R}^{N \times N}$. It has been derived that

$$\nabla F(M) = -2a \times \left[ \sum_m \Gamma_m \left( h^m \right)^{\circ} \mid [\bar{Z} - Z] \odot Z \right.$$  

$$\odot \left( (1_{N \times N} - Z) \odot \left[ (h^m)^{\circ} \otimes M \right] \right] \right) - 2a$$

Downloaded From: http://nanolithography.spiedigitallibrary.org/ on 04/08/2014 Terms of Use: http://spiedl.org/terms
\[
\sum_m \Gamma_m(h^m)^* \otimes [(\bar{Z} - Z) \otimes (1_{N \times N} - Z) \otimes (h^m \otimes M)]
\]

(27)

where \(\otimes\) is the element-by-element multiplication operator, \((h^m)^*\) rotates \(h^m\) by 180° in both row and column directions, \(*\) is the conjugate operation, and \(1_{N \times N} \in \mathbb{R}^{N \times N}\) has all entries equal to 1. On the other hand,

\[
\frac{\partial S}{\partial m_j} = -\frac{1}{12} N^2 \sum_{k=1}^{N^2} \left[ g_{kj} \left( N^2 \sum_{j=1}^{N^2} g_{kj} m_j - 2 \right) \left( \sum_{j=1}^{N^2} g_{kj} m_j - 4 \right) \left( \sum_{j=1}^{N^2} 2g_{kj} m_j - 3 \right) \right]
\]

\[
-\frac{1}{12} N^2 \sum_{k=1}^{N^2} \left[ g_{kj} \left( \sum_{j=1}^{N^2} g_{kj} m_j \right) \left( \sum_{j=1}^{N^2} g_{kj} m_j - 2 \right) \left( \sum_{j=1}^{N^2} 2g_{kj} m_j - 3 \right) \right]
\]

\[
-\frac{1}{12} N^2 \sum_{k=1}^{N^2} \left[ g_{kj} \left( \sum_{j=1}^{N^2} g_{kj} m_j \right) \left( \sum_{j=1}^{N^2} g_{kj} m_j - 2 \right) \left( \sum_{j=1}^{N^2} 2g_{kj} m_j - 3 \right) \right]
\]

\[
-\frac{1}{6} N^2 \sum_{k=1}^{N^2} \left[ g_{kj} \left( \sum_{j=1}^{N^2} g_{kj} m_j \right) \left( \sum_{j=1}^{N^2} g_{kj} m_j - 2 \right) \left( \sum_{j=1}^{N^2} g_{kj} m_j - 4 \right) \right]
\]

(28)

Thus,

\[
\nabla S(M) = -\frac{1}{12} \mathbf{g}^* \otimes [(G_M - 2_{N \times N}) \otimes (G_M - 4_{N \times N}) \otimes (2G_M - 3_{N \times N})]
\]

\[
-\frac{1}{12} \mathbf{g}^* \otimes [G_M \otimes (G_M - 4_{N \times N}) \otimes (2G_M - 3_{N \times N})]
\]

\[
-\frac{1}{12} \mathbf{g}^* \otimes [G_M \otimes (G_M - 2_{N \times N}) \otimes (2G_M - 3_{N \times N})]
\]

\[
-\frac{1}{6} \mathbf{g}^* \otimes [G_M \otimes (G_M - 2_{N \times N}) \otimes (G_M - 4_{N \times N})]
\]

(29)

where \(\mathbf{g}^*\) rotates \(\mathbf{g}\) by 180° in both row and column directions, \(G_M\) is represented in Eq. (20), and \(\mathbf{n}_{N \times N} \in \mathbb{R}^{N \times N}\) has all entries equal to \(n\).

6 Gradient-based PBOPC Optimization Algorithm with Improved Manufacturability

In this section, a PBOPC optimization algorithm is proposed to simultaneously improve the fidelity of the print image and reduce the rectangle count of the fractured mask pattern. First, the mask pattern is initialized as the desired pattern. In addition, the upper bound of the rectangle count \(L\) is set by the user. Subsequently, the cost sensitivity is calculated to navigate the cost function in the descent direction during the optimization process. The pixel \(p(x_{\text{max}}, y_{\text{max}})\) with the largest absolute value of cost sensitivity is found and assigned with the highest priority to be flipped. Then, we check if \(p(x_{\text{max}}, y_{\text{max}})\) belongs to the changeable pixels, which will be defined shortly. According to Section 4.2, the opposite vertical angle is not allowable in the mask pattern. Thus, any pixel flip that introduces opposite vertical angle is prohibited. We define the changeable pixel as the pixel, flip of which does not introduce any opposite vertical angle. If \(p(x_{\text{max}}, y_{\text{max}})\) is a changeable pixel, we flip it according to the sign of its cost sensitivity. If the pixel value is 0 and the cost sensitivity is negative, then the pixel is flipped to 1. If the pixel value is 1 and the cost sensitivity is positive, then the pixel is flipped to 0. After that, we update the fidelity cost term \(F\) and the rectangle count in the fractured mask pattern. The value of the flipped pixel is accepted only if the fidelity cost function is not increased, and the rectangle count is under the prescribed upper bound \(L\). Otherwise, the flipped pixel will be restored to its original value. Finally, the algorithms are terminated whenever no pixel can be flipped. The proposed algorithm is summarized in the following:

1. **Step 1**: Initialize the mask pattern and the upper bound of rectangle count, such that \(M = \bar{Z}\), and \(#\text{(rectangle)}\) \(\leq L\).

2. **Step 2**: Calculate the cost sensitivity \(\nabla D(M)\) [see Eqs. (26)–(29)].

3. **Step 3**: Find the pixel \(p(x_{\text{max}}, y_{\text{max}})\) in the mask pattern, which has the maximum absolute value of cost sensitivity, i.e., \(|\nabla D(p(x_{\text{max}}, y_{\text{max}}))| = \max(|\nabla D|)\).

4. **Step 4**: If \(p(x_{\text{max}}, y_{\text{max}})\) \(\in\) changeable pixels 

   Go to Step 5.
Otherwise

Go to Step 7.

Step 5: Update the value of \( p(x_{\text{max}}, y_{\text{max}}) \) according to the sign of \( \nabla D(p(x_{\text{max}}, y_{\text{max}})) \):

If \( p(x_{\text{max}}, y_{\text{max}}) = 0 \) and \( \nabla D(p(x_{\text{max}}, y_{\text{max}})) < 0 \)

\( p(x_{\text{max}}, y_{\text{max}}) = 1 \).

Else If \( p(x_{\text{max}}, y_{\text{max}}) = 1 \) and \( \nabla D(p(x_{\text{max}}, y_{\text{max}})) > 0 \)

\( p(x_{\text{max}}, y_{\text{max}}) = 0 \).

Step 6: If \( p(x_{\text{max}}, y_{\text{max}}) \) is flipped in Step 5, then we check it to determine whether accepting the flip operation:

If (fidelity cost term \( F \) is increased) or (#(rectangle) > \( L \))

Restore \( p(x_{\text{max}}, y_{\text{max}}) \) to its original value.

Step 7: Mark the location of \( p(x_{\text{max}}, y_{\text{max}}) \) in the cost sensitivity as a transversed pixel, i.e., \( \nabla D(p(x_{\text{max}}, y_{\text{max}})) = 0 \).

Step 8: If \( \nabla D \neq 0 \)

Go to Step 3.

Otherwise

If no pixel is flipped in the current iteration

End.

Otherwise

Go to Step 2.

In the proposed algorithm, whenever any pixel is flipped, we will recalculate the aerial image and check whether the fidelity cost term is increased. It is known that the aerial image calculation is computational complex. Thus, the speed of the proposed algorithm critically depends on the ability to quickly recalculate the aerial image when one pixel changes. To achieve this goal, the electric field caching technique (EFCT) is applied to speed up the aerial image recalculation.\(^{32}\) Initially, we calculate and save every electric field component \( E^m \) in the partially coherent system. According to Eq. (4), we have

\[
E^m(x, y) = M(x, y) \otimes h^m(x, y). \tag{30}
\]

If a pixel located at \((x_0, y_0)\) is flipped from 0 to 1, the electric field components are updated as

\[
E^m(x, y) = E^m(x, y) + h^m(x - x_0, y - y_0). \tag{31}
\]

If a pixel located at \((x_0, y_0)\) is flipped from 1 to 0, the electric field components are updated as

\[
E^m(x, y) = E^m(x, y) - h^m(x - x_0, y - y_0). \tag{32}
\]

Whenever the electric field components are updated, the aerial image can be recalculated as

\[
I(x, y) = \sum_m |\Gamma_m| E^m(x, y)^2. \tag{33}
\]

Since each time we only flip one pixel, the EFCT can quickly update the aerial image. Thus, the computational intensity incurred by the aerial image calculation is effectively alleviated. However, one disadvantage of the EFCT is that the aerial images calculated by EFCT and by Eq. (4) are not perfectly equal to each other. This difference may be caused by the rounding errors in MATLAB™ software. Although the difference is very small, error accumulation will become pronounced after a number of iterations. In order to solve this problem, we reset the electric field components using Eq. (30) whenever we have applied EFCT for ten times.

### 7 Simulations

A simulation of the proposed algorithm is shown in Fig. 6, where black and white represent 0 and 1, respectively. In this simulation, the desired pattern is two-bars with dimension of 450 nm × 450 nm as illustrated in Fig. 6(a). The pixel size is 5.625 nm × 5.625 nm. The width of each bar is 95.625 nm. The distance between the two bars is 33.75 nm. The convolution kernel is shown in Eq. (8) with NA = 1.25 and \( \lambda = 193 \) nm, and assumed to vanish outside the area \( A_b \) defined by \( x, y \in [-56.25 \text{ nm}, 56.25 \text{ nm}] \). The illumination is an annular illumination with \( \sigma_{\text{inner}} = 0.3 \) and \( \sigma_{\text{outer}} = 0.4 \).

![Fig. 6 Performance comparison between the steepest descent algorithm and the proposed algorithm. Top row (from left to right) shows: (a) the desired pattern, (b) the optimized mask using the steepest descent algorithm, and (c) the optimized mask using the proposed algorithm. Bottom row shows the output patterns corresponding to the input masks in the top row. \( \sigma_{\text{inner}} = 0.3 \) and \( \sigma_{\text{outer}} = 0.4 \). Black and white represent 0 and 1, respectively.](image-url)
In Fig. 6, we compare the optimized masks obtained from the steepest descent PBOPC algorithm\textsuperscript{17} and the proposed algorithm. Figure 6(d) is the the output pattern corresponding to the desired pattern with a pattern error of 400. The pattern error is defined as the square of the \L2 norm of the difference between the output pattern and the desired pattern. Figure 6(b) is the optimized mask pattern using the steepest descent algorithm with 120 rectangles in the fractured pattern. The runtime is 74 s. The computation hereafter was done on an Intel(R) Pentium(R) 4 CPU, 3.00 GHz, 0.99 GB of RAM. The computation platform is MATLAB\textsuperscript{TM} software. The corresponding output pattern is shown in Fig. 6(e) with a pattern error of 48. For the steepest descent algorithm, the step length is 2. In Eq. (11), \(a = 25\) and \(\varepsilon = 0.3\). In addition, we applied the quadratic penalty and wavelet penalty to the steepest descent algorithm with both weights equal to 0.01.\textsuperscript{17,33} The formulas of the quadratic penalty and the wavelet penalty are summarized in the Appendix. Figure 6(c) is the optimized mask pattern using the proposed algorithm with 100 rectangles in the fractured pattern. For the proposed algorithm, the weight of the manufacturability cost term \(\gamma = 0.5\) in Eq. (25). The upper bound of rectangle count \(L = 100\) in Step 1 of the proposed algorithm is described in Sec. 6. The runtime is 387 s. Compared to the steepest descent algorithm, the proposed algorithm reduces the rectangle count by 16.7\%, thus improving the manufacturability of the mask. It is noted that Fig. 6(c) is asymmetric, because the proposed algorithm flips every pixel on the mask independently. In order to obtain symmetric structures, the proposed algorithm can be easily modified to just optimize the left half part of the mask pattern with respect to the midline. The symmetric pixels in the right half part are flipped in the same way. The output pattern corresponding to Fig. 6(c) is shown in Fig. 6(f) with a pattern error of 27. Thus, the proposed algorithm reduces the output pattern error by 43.8\%, consequently resulting in greater pattern fidelity than the steepest descent algorithm. However, the proposed algorithm increases the runtime by a factor of 4.

In our proposed algorithm, the weight of the manufacturability cost term \(\gamma\) and the upper bound of rectangle count \(L\) are the two key parameters that influence the pattern error and the rectangle count in the fractured mask pattern. These two parameters can be determined by a simple line search method. In this method, we apply uniform sampling for each parameter and evaluate pattern errors for each parameter value. Then, we adopt the optimal value of the parameter. The analytical approaches to design these parameters fall outside the scope of our paper and are topics for future work. We first investigate the feasible values of \(\gamma\). Given \(L = 100\), the relationship between the values of \(\gamma\) and the resulting pattern errors is indicated by the black solid line in Fig. 7. According to this relationship, \(\gamma = 0.5\) is selected, since it leads to the minimum pattern error of 27. Next, we investigate the feasible values of \(L\). Given \(\gamma = 0.5\), the relationship between the values of \(L\) and the resulting pattern errors is indicated by the black solid line in Fig. 8. Figure 9 illustrates the simulation results obtained from the proposed algorithm with different values of \(L\). Figures 9(a)–9(c) illustrate the optimized mask patterns using \(L = 20, 70,\) and 100. Figures 9(d)–9(f) illustrate the corresponding output patterns. It is observed that the parameter \(L\) is effective to control the complexity of the mask pattern. In addition, the optimization with larger \(L\) is more likely to result in less pattern error. That is because larger \(L\) allows more fractured rectangles in the mask, thus increasing the degree of freedom during the optimization process. According to the above analysis, we assign \(L = 100\), since it leads to 20 less rectangles than the steepest descent algorithm, and results in the minimum pattern error of 27.

In order to prove the universality of the proposed algorithm, another simulation is provided in Fig. 10. The desired pattern illustrated in Fig. 10(a) is a U-shape with dimension of 562.5 nm \(\times\) 562.5 nm. The line width and the distances between lines are 67.5 nm. For the steepest descent algorithm, all parameters are the same as Fig. 6. For the proposed algorithm, the weight of the manufacturability cost term is \(\gamma = 0.1\), and the upper bound of the rectangle count is \(L = 140\). Compared to the steepest descent algorithm, the proposed algorithm reduces the output pattern error by 29.0\%, and reduces the rectangle count by 12.5\%. That means that the proposed algorithm simultaneously improves the fidelity of the printed image and the

![Fig. 7](image-url) The relationship between the weight of the manufacturability cost term \(\gamma\) and the resulting pattern error.

![Fig. 8](image-url) The relationship between the upper bound of rectangle count \(L\) and the resulting pattern error.
Fig. 9 Performance comparison of the proposed algorithms with different values of $L$. Top row (from left to right) shows: (a) the optimized mask using $L = 20$, (b) the optimized mask using $L = 70$, and (c) the optimized mask using $L = 100$. Bottom row shows the output patterns corresponding to the input masks in the top row. $\sigma_{\text{inner}} = 0.3$ and $\sigma_{\text{outer}} = 0.4$. Black and white represent 0 and 1, respectively.

manufacturability of the mask. On the other hand, the steepest descent algorithm costs 107 s, while the proposed algorithm costs 1619 s. Therefore, the proposed algorithm increases the runtime by a factor of 14. The line search procedure of $\gamma$ and $L$ are discussed next. Given $L = 140$, the relationship between the values of $\gamma$ and the resulting pattern errors is indicated by the blue dashed line in Fig. 7. According to this relationship, $\gamma = 0.1$ is selected, since it leads to the minimum pattern error of 44. In addition, given $\gamma = 0.1$ the relationship between the values of $L$ and the resulting pattern error.

Fig. 10 Performance comparison between the steepest descent algorithm and the proposed algorithm. Top row (from left to right) shows: (a) the desired pattern, (b) the optimized mask using the steepest descent algorithm, and (c) the optimized mask using the proposed algorithm. Bottom row shows the output patterns corresponding to the input masks in the top row. $\sigma_{\text{inner}} = 0.3$ and $\sigma_{\text{outer}} = 0.4$. Black and white represent 0 and 1, respectively.
errors is indicated by the blue dashed line in Fig. 8. Figure 11 illustrates the simulation results obtained from the proposed algorithm with different values of \( L \). Figures 11(a)–11(c) illustrate the optimized mask patterns using \( L = 20 \), 80, and 140. Figures 11(d)–11(f) illustrate the corresponding output patterns. According to the above simulations, we assign \( L = 100 \), since it leads to 20 less rectangles and a lower pattern error than the steepest descent algorithm.

As shown in the above simulations, using the line search method to properly choose the parameters \( y \) and \( L \), the proposed algorithm can lead to greater resolution of the printed image and less rectangle count in the fractured mask pattern than the steepest descent PBOPC algorithm. Therefore, the proposed algorithm can simultaneously enhance the resolution of the optical lithography system and improve the manufacturability of the optimized mask pattern.

8 Conclusion and Future Work

This paper proposed a gradient-based PBOPC optimization framework to simultaneously improve the resolution of the optical lithography system and the manufacturability of the mask pattern. The partially coherent imaging system is modeled by the Fourier series expansion model. The topography filter and the manufacturability cost term are applied to formulate and control the rectangle count in the fractured mask pattern during the optimization process. Cost sensitivity is used to drive both the imaging pattern error and the rectangle count in the descent directions. The influence of the user-defined parameters on the performance of the proposed algorithm is discussed. Simulations illustrate that using the line search method to properly choose the parameters, the proposed approaches are effective to enhance the resolution of the optical lithography system and improve the manufacturability of the optimized mask pattern.

The simulations in this paper use simple patterns, such as two-bars and U-shape. However, the layouts of the current IC technologies include more complex Manhattan geometries. Hence, in future work, we plan to test the proposed algorithms using various complex patterns captured from the real IC layouts. In addition, the optimized mask patterns obtained from the proposed algorithms include numerous slivers which increase the inaccuracy of the mask fabrication. Thus, another important direction of our future work is to improve the current algorithms to effectively remove the slivers in the optimized mask patterns.

Appendix

In the PBOPC optimization based on the steepest descent method, we adopt the parametric transformation as follows\(^\text{16}\)

\[
m_{ij} = \frac{1 + \cos(\theta_{ij})}{2}, \quad j = 1, \ldots, N^2, \tag{34}\]

The quadratic penalty term is \( R_Q(m) = 4m^T(1 - m) \). For each pixel value, the corresponding penalty is the quadratic function \( r(m_i) = 1 - (2m_i - 1)^2, \ i = 1, \ldots, N^2 \). The gradient of \( R_Q \) is \( \nabla R_Q(\theta) = -\frac{1}{4}(-8m + 4) \cdot \sin(\theta) \).

The wavelet penalty term is

\[
R_W(m) = h_{11}^2 + h_{12}^2 \cdots + h_{N/2, N/2}^2 + v_{11}^2 + v_{12}^2 \cdots + v_{N/2, N/2}^2 + d_{11}^2 + d_{12}^2 \cdots + d_{N/2, N/2}^2. \tag{35}\]

where

\[
h_{ij} = m_{[2j-1]+1}[2j-1+1][2j-1+1] - m_{[2j-1]+2}[2j-1+2][2j-1+2] + m_{[2j-1]+1}[2j-1+2][2j-1+2] - m_{[2j-1]+2}[2j-1+1][2j-1+1], \tag{36}\]
\[ v_{ij} = \frac{m_1[(i-1)+1][j-1]+m_2[i-1]+1]}{2[i-2]+1} + \frac{m_3[i-1]2+[j-1]+1}{2[i-2]+1} + \frac{m_4[i-1]+1}{2[i-2]+1} \]  
(37)

\[ d_{ij} = \frac{m_1[(i-1)+1][j-1]+1}{2[i-2]+1} - \frac{m_2[i-1]+1}{2[i-2]+1} + \frac{m_4[i-1]+1}{2[i-2]+1} \]  
(38)

for \( i, j = 1, \ldots, N/2 \). The gradient of the wavelet penalty is given as

\[ \frac{\partial R_W}{\partial q_{i-j}} = \frac{1}{2} \sin^{(2)}[(2i+1)p]2(j-1)+q] \times (3m_2[i-1]+p)[j-1]+q] - m_2(i-1)+p]}{2[i-2]+1} + \frac{m_4[i-1]+1}{2[i-2]+1} \]  
(39)

where \( i, j = 1, \ldots, N/2 \), \( p, q = 10r+2 \), \( p_1 = (p+1)\mod 2 \) and \( q_1 = (q+1)\mod 2 \).

Acknowledgments

We thank the financial support by Key Program of National Natural Science Foundation of China under Grant No. 6170538003, National Science and Technology Major Project, Program of Education Ministry for Changjiang Scholars in University, Excellent Young Scholars Research Foundation of Beijing Institute of Technology (Grant No. 2010Y0408), and the Basic Research Foundation of Beijing Institute of Technology (Grant No. 20100442029). Our work is also supported by Program for New Century Excellent Talents in University (NCET).

References