Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 589128, 9 pages
doi:10.1155/2012/589128

Research Article

Line Search-Based Inverse Lithography Technique for Mask Design

Xin Zhao and Chris Chu

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA

Correspondence should be addressed to Xin Zhao, xinzhao@iastate.edu

Received 24 March 2012; Accepted 3 August 2012

Academic Editor: Gi-Joon Nam

Copyright © 2012 X. Zhao and C. Chu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As feature size is much smaller than the wavelength of illumination source of lithography equipments, resolution enhancement technology (RET) has been increasingly relied upon to minimize image distortions. In advanced process nodes, pixelated mask becomes essential for RET to achieve an acceptable resolution. In this paper, we investigate the problem of pixelated binary mask design in a partially coherent imaging system. Similar to previous approaches, the mask design problem is formulated as a nonlinear program and is solved by gradient-based search. Our contributions are four novel techniques to achieve significantly better image quality. First, to transform the original bound-constrained formulation to an unconstrained optimization problem, we propose a new noncyclic transformation of mask variables to replace the wellknown cyclic one. As our transformation is monotonic, it enables a better control in flipping pixels. Second, based on this new transformation, we propose a highly efficient line search-based heuristic technique to solve the resulting unconstrained optimization. Third, to simplify the optimization, instead of using discretization regularization penalty technique, we directly round the optimized gray mask into binary mask for pattern error evaluation. Forth, we introduce a jump technique in order to jump out of local minimum and continue the search.

1. Introduction

For modern very large-scale integration (VLSI) design, the traditional VLSI physical design problems (e.g., floorplanning [1–3], clustering [4, 5], placement [6], and routing) used to play the critical role on coping with the ever-increasing design complexity. However, as semiconductor manufacturers move to advanced process nodes (especially 45 nm process and below), lithography has become a greater challenge due to the fundamental constraints of optical physics. Because feature size is much smaller than the wavelength of illumination source (currently 193 nm), the image formed on wafer surface is distorted more and more seriously due to optical diffraction and interference phenomena. The industry has been investigating various alternatives (e.g., EUV lithography, E-beam lithography), but none of them is ready in the foreseeable future. As a result, semiconductor manufacturers have no choice but to keep using the existing equipments in patterning the progressively smaller features.

Given the limitation of lithography equipments, resolution enhancement technology (RET) such as optical proximity correction (OPC), phase shift mask (PSM), and double patterning has been increasingly relied upon to minimize image distortions [7]. In recent years, pixelated mask, which allows great flexibility in the mask pattern, has become essential for RET to achieve better resolution.

For the design of pixelated mask, the most popular and successful approach is to formulate it as a mathematical program and solve it by gradient-based search [8–14]. Granik [8] considered a constrained nonlinear formulation. Poonawala and Milanfar [9, 14, 15] proposed an unconstrained nonlinear formulation, and employed a regularization framework to control the tone and complexity of the synthesized masks. Ma and Arce [11, 16] presented a similar unconstrained nonlinear formulation targeting PSM. Ma and Arce [12, 16] focused on partially coherent illumination and used singular value decomposition to expand the partially coherent imaging equation by eigenfunctions into a sum of coherent systems (SOCSs). All works discussed above

utilized the steepest descent method to solve the nonlinear programs. Ma and Arce [10] demonstrated that the conjugate gradient method is more efficient. The work of Yu and Pan [17] is an exception to the mathematical programming approach Instead, a model-based pixel flipping heuristic is proposed.

In this paper, we focus on the design of pixelated binary mask in a partially coherent imaging system (the techniques proposed in this paper can all be easily extended to PSM and other imaging systems). Similar to previous approaches, we formulate the problem as an unconstrained nonlinear program and solve it by iterative gradient-based search. The main contributions of this paper are listed below.

(i) To transform the problem formulation from a bounded optimization to an unconstrained one, we propose a new noncyclic transformation of mask variables to replace the widely used cyclic one. Our transformation is monotonic and allows a better control of flipping pixels.

(ii) Based on this new transformation, we present a highly efficient line search-based technique to solve the resulting unconstrained optimization. Because of the non-cyclic nature of the transformation, the solution space is not so rugged. Therefore, our algorithm can find much better binary masks for the inverse lithography problem.

(iii) A jump technique: as gradient-based search techniques will be trapped at a local minimum, we introduce a new technique named *jump* in order to jump out of the local minimum and continue the search.

(iv) We apply a direct rounding technique to regularize gray masks into binary ones instead of adding a discretization regularization penalty to the cost function as in [14] and [16]. This simplifies the computation and achieves better results as the experiment results show.

The rest of this paper is organized as follows. The formulation of the inverse lithography problem is explained in Section 3. Section 4 describes in details the flow of our algorithm and the four novel techniques that we proposed. Section 5 presents the experimental results. The paper is concluded in Section 6.

2. New Algorithmic Technique Used

The inverse lithography technique for mask design has been proposed in [8, 15] in 2006 and has been widely discussed in recent years as semiconductor manufacturers move to advanced process nodes. But so far, there is not an effective search method proposed because of the complicated solution space of this problem. We introduce a novel transformation for mask pixel, which enables an effective line search technique.

3. Problem Formulation

In an optical lithography system, a photo mask is projected to a silicon wafer through an optical lens. An aerial image of the mask is then formed on the wafer surface, which is covered by photoresist. After developing and etching, a pattern similar to the one on the mask is formed on the wafer surface. To simulate the pattern formation on the wafer surface for a given mask, we first describe below a projection optics model and a photoresist model. After that, we present the formulation of the mask design problem.

3.1. Projection Optics Model. The Hopkins diffraction model [13] is widely used to approximate partially coherent optics systems. To reduce the computational complexity of the Hopkins diffraction model, the Fourier series expansion model [18] is a common approach. In this paper, we followed this model.

The Fourier series expansion model approximates the partially coherent imaging system as a sum of coherent system (SOCS). Based on this model, the computation of the aerial image I of a pixelated mask M is given in (1) and illustrated in Figure 1. Here, the dimensions of the pixelated mask and the image are $m \times n$. The illumination source is partitioned into $N \times N$ sources. u and are the Fourier series coefficients and spatial kernels, respectively:

$$I(M) = \sum_{p=1}^{N\cdot N} u_p \left| h_p \otimes M \right|^2. \tag{1}$$

Note that the convolution $h \otimes M$ can be achieved in frequency domain using fast Fourier transform $\mathcal{FFT}$ and inverse fast Fourier transform $\mathcal{FFT}^{-1}$ as shown in the following:

$$I(M) = \sum_{p=1}^{N\cdot N} u_p \left| \mathcal{FFT}^{-1}\left\{ \mathcal{FFT}\left(h_p\right) \cdot \mathcal{FFT}(M) \right\} \right|^2. \tag{2}$$

3.2. Photoresist Model. To model the reaction of the photoresist to the intensity of light projected on it, we use the constant threshold model as follows

$$z_i = \begin{cases} 0 & \text{if } I_i \le t_r, \\ 1 & \text{if } I_i > t_r, \end{cases} \tag{3}$$

where I_i and z_i are the light intensity and the corresponding reaction result of the photoresist at pixel i on the wafer surface, respectively, and t_r is the threshold of the photoresist.

Thus, the pattern z formed on the wafer surface can be expressed as a function of the mask M based on (2) and (3). In order to make z differentiable so that gradient-based search can be applied, we approximate the above constant threshold model with the sigmoid function

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-a(x-v)}}, \tag{4}$$

where the parameter a determines the steepness of the sigmoid function around $x = v$. The larger value of a is,

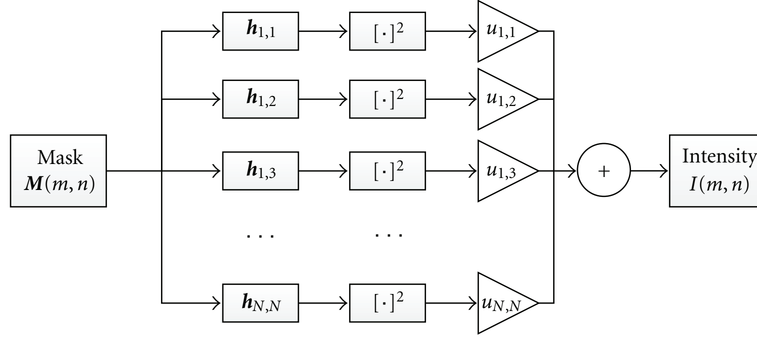


Figure 1: Fourier series expansion model for partially coherent system.

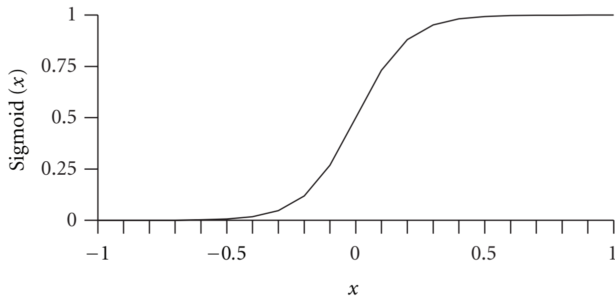


Figure 2: The sigmoid function with $a = 10$ and $v = 0$.

the steeper and hence the closer to the constant threshold model the sigmoid function will be. The sigmoid function with $a = 10$, $v = 0$ is illustrated in Figure 2.

Using the sigmoid function, the reaction of the photoresist at pixel i for a mask M is

$$z_i(M) \approx \frac{1}{1 + e^{-a(I_i(M)-t_r)}}. \quad (5)$$

3.3. Our Inverse Lithography Problem Formulation. Inverse lithography treats mask design as an inverse problem of imaging. Given a target pattern $\hat{z}$, the problem is to find a mask M^* such that the corresponding pattern $z(M^*)$ on the wafer surface is as close to $\hat{z}$ as possible [19].

The error between the target pattern $\hat{z}$ and the generated pattern $z(M)$ for any mask M is commonly defined as

$$E(M) = \sum_{i=1}^{m\cdot n} (\hat{z}_i - z_i(M))^2. \quad (6)$$

So the inverse lithography problem is formulated as

$$M^* = \arg\min_{\forall i, M_i \in \{0,1\}} E(M). \quad (7)$$

Combining (5) and (6) with (7), the problem is written as

$$M^* = \arg\min_{\forall i, M_i \in \{0,1\}} \sum_{i=1}^{m\cdot n} \left(\hat{z}_i - \frac{1}{1 + e^{-a(I_i(M)-t_r)}}\right)^2, \quad (8)$$

where $I_i(M)$ is the light intensity at pixel location i calculated by (2).

4. Line Search-Based Inverse Lithography Technique

As the value of each pixel M_i should be 0 or 1, the inverse lithography problem is an integer nonlinear program. To make it easier to solve, a common approach is to relax the integer constraints to $0 \leq M_i \leq 1$ for all i [8–12, 14]. Therefore, the problem becomes a bounded non-linear program. To further simplify the program, it is also common to convert it into an unconstrained non-linear program [8–12, 14]. It is achieved by a transformation $M_i = T(\beta_i)$ which maps an unbounded variable β_i into the range $[0, 1]$. (We will discuss this transformation in Section 4.1.) The program is then solved with respect to β's domain.

This unconstrained non-linear program can be solved by an iterative gradient-based search method. Starting from some point β in the solution space, a search direction, which can be decided based on the gradient of (6), is first found. Then a step of a certain size along the search direction is taken. Thus, a new point, which hopefully has less pattern error, is reached. The search is repeated until the error cannot be further reduced.

In this paper, we apply this iterative gradient-based search method, which is outlined in Algorithm 1. Our contributions are four novel techniques as described in Sections 4.1, 4.2, 4.3, and 4.4 to reduce pattern error over previous works.

In particular, we use the steepest descent method, that is, the search direction is the negative of the local gradient of (6). But our techniques are not limited only to the steepest descent method. It should be applicable to other iterative gradient-based search approaches like conjugate gradient method.

4.1. Novel Transformation for Mask Pixel. As explained above, to convert the inverse lithography problem into an unconstrained optimization problem, we need a transformation $T : \mathcal{R} \to [0, 1]$. Then we can use an unbounded variable β_i to represent each pixel based on $M_i = T(\beta_i)$.

One such transformation is proposed by Poonawala and Milanfar [14]:

$$M_i = \frac{1 + \cos\beta_i}{2}. \quad (9)$$

(1) Transform initial mask into β // Section 4.1
(2) Repeat
(3) Find the search direction d at β // Equation (12)
(4) Determine the step size S // Section 4.2
(5) $\beta^{\text{new}} = \beta + S * d$
(6) Generate *gray* mask $M = T(\beta^{\text{new}})$ // Equation (11)
(7) // Round M to binary as described in Section 4.4
(8) Evaluate pattern error $E(M)$ // Equations (2), (3), and (6)
(9) $\beta = \beta^{\text{new}}$
(10) Until pattern error is not improving

Algorithm 1: Generic framework for iterative gradient-based search. Note that specific details about our algorithm are given in the comments.

This idea is widely adopted by later works [9–12]. We call it the cosine transformation.

In gradient-based search, a line search along the search direction is typically performed to determine the step size S to get to a local minimum (step 4 in Algorithm 1). The line search will be more effective if the function $E(M)$ along the search direction is smooth and, better yet, convex. However, the cosine transformation is a cyclic function. It is clearly not a one-to-one transformation. By increasing the value of β_i, M_i changes its value between 0 and 1 periodically. As a result, when β is moving along any direction, $E(\mathrm{M})$ may keep jumping up and down as M_i keeps switching between 0 and 1.

To illustrate this, we consider the algorithm described in Chapter 7 of Ma and Arce [16], which solved the same problem formulation as our paper. It also applied the steepest descent method, but it used the cosine transformation. The pattern error function (6) is turned into the following:

$$E(\beta) = \sum_{i=1}^{m\cdot n}\left(\hat{z}_i - \frac{1}{1+e^{-a(\sum_{p=1}^{N\cdot N} u_p |h_p \otimes ((1+\cos\beta)/2)|^2 - t_r)}}\right)^2. \tag{10}$$

Using the software code and the target pattern (as shown in Figure 3) provided by [16], when β is moved along the negative gradient direction of (10), the function $E(\beta)$ is illustrated in Figures 4 and 5. It shows that the function changes in a very chaotic manner. We have observed a large number of experiments on different target patterns and different current masks. The function $E(\beta)$ always shows a similar chaotic behavior. It makes line search very difficult. In theory, the negative gradient points out the direction for each pixel to be adjusted to achieve the minimal value of pattern error. However, the gradient only provides the direction of change at the local point. Because of the cyclic property of (9), the pixels on the mask may be flipped to the wrong direction if the step size S is not set appropriately. This makes the gradient-based search method very ineffective. In fact, the common practice in previous works [9–12, 14] is to set the step size to some fine-tuned constant instead of computed-by-line search.

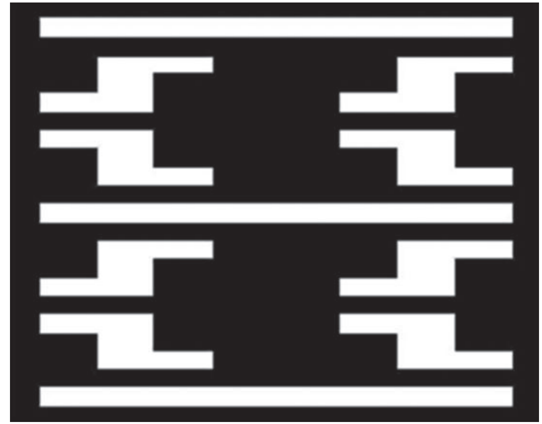


Figure 3: Target pattern from [16] with 184×184 pixels.

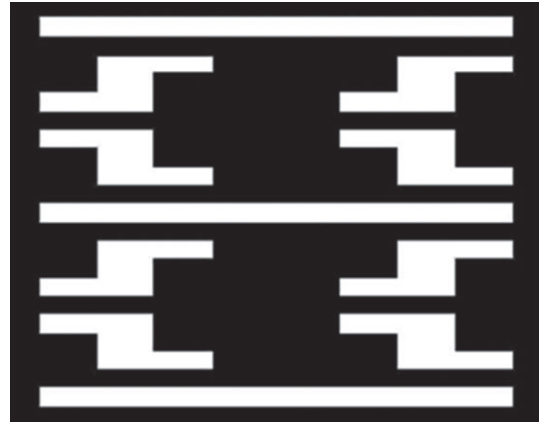


Figure 4: Pattern error based on cosine transformation.

We propose a new transformation for M_i based on the sigmoid function (see (4)):

$$M_i = \frac{1}{1+e^{-A(\beta_i - T_R)}}, \tag{11}$$

where A is the steepness control parameter and T_R specifies the transition point of the function. A larger A will cause the pixel values to be closer to 0 or 1. T_R can be set to any value and is set to 0 in this paper. We call this the sigmoid

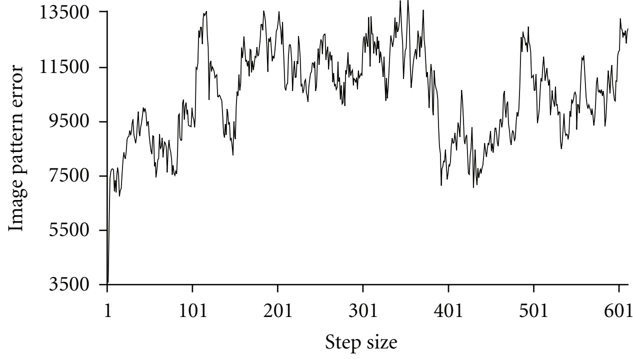


Figure 5: Enlarged version of Figure 4 with step size from = 0 to 25.

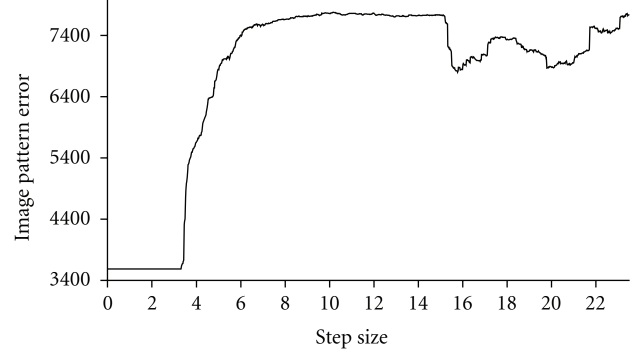


Figure 6: Pattern error based on sigmoid transformation.

transformation. As the sigmoid transformation is a strictly increasing function, when β is moved along any direction, each mask pixel is flipped at most once.

Based on the sigmoid transformation, the gradient of (6) is

$$\frac{\partial E(\beta)}{\partial \beta} = -a \cdot A \left\{ \sum_{p=1}^{N \cdot N} u_p \left[\left(h_p \otimes M \right) \odot (\hat{z} - z) \odot z \odot (1 - z) \right] \otimes h_p^{*T} \right\} \odot M \odot (1 - M) - a \cdot A \cdot \left\{ \sum_{p=1}^{N \cdot N} u_p \left[\left(h_p^{*} \otimes M \right) \odot (\hat{z} - z) \odot z \odot (1 - z) \right] \otimes h_p^{T} \right\} \odot M \odot (1 - M), \quad (12)$$

where $1 = [1, \ldots, 1]^T \in \mathcal{R}^{m \times n}$, $\odot$ is the element-by-element multiplication operator, and h_p^* is the conjugate of h_p. We have performed a large number of experiments on different target patterns and different current masks. When β is moved along the negative gradient direction, the function $E(\beta)$ is almost always unimodal. One typical example is shown in Figures 6 and 7. (Note that not every pixel can be flipped along the negative gradient direction, as we will explain in Section 4.2.) This makes it feasible to apply line search to heuristically minimize the pattern error. Note that gradient calculation is very expensive due to the four convolutions in (12). Hence, once a gradient is calculated, it is desirable to perform line search to minimize the pattern error as much as possible in order to reduce the number of iterations (i.e., gradient calculations) of the gradient-based search algorithm. As Figure 7 shows, by performing line search along the negative gradient direction, the image pattern error can be effectively reduced from around 3600 to below 3100 in one iteration.

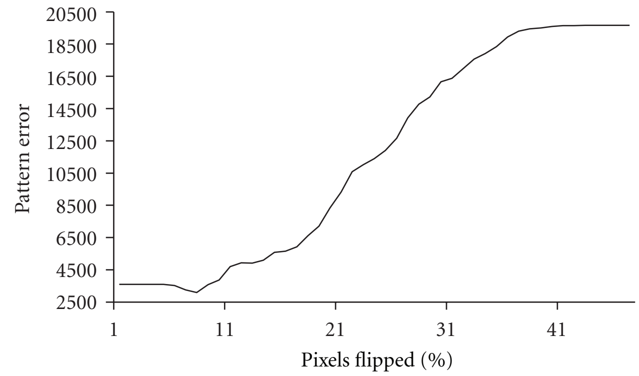


Figure 7: Enlarged version of Figure 6 with percentage from 5.1% to 9.0%.

4.2. Highly Efficient Line Search Technique. In this section, we present a highly efficient line search technique to determine the step size in step 4 in Algorithm 1 to minimize pattern error. We observe that in each iteration, the shape of the function $E(\beta)$ along the direction of the negative gradient is almost always like the curve shown in Figure 6. We employ golden section method for line search. Golden section search is an iterative technique which successively narrows the search range.

Because the final optimized mask should be a binary one, we need to round the gray mask, which is given by (11), to binary according to some rounding threshold t_m. In other words,

$$M_i^{\text{binary}}(\beta_i) = \begin{cases} 0, & M_i(\beta_i) < t_m, \\ 1, & M_i(\beta_i) \geq t_m, \end{cases} \quad (13)$$

where M^{binary} is the resulting binary mask. Here, we simply set t_m to 0.5.

When moving along the negative gradient direction, as the value of each pixel M_i is changed monotonically due to our new transformation, we can easily control the number of pixels flipped (i.e., changed from below t_m to above t_m or vice versa) during line search. This idea is explained below.

Given the current mask specified by β and the negative gradient direction d, (11) can be written as a function of S as

$$M_i(S) = \frac{1}{1 + e^{-A(\beta_i - S \times d_i - T_R)}}. \quad (14)$$

By substituting (14) into (13) and rearranging, we get the following

if $d_i \geq 0$,

$$M_i^{\text{binary}}(S) = \begin{cases} 0, & S > S_i, \\ 1, & S \leq S_i, \end{cases} \quad (15)$$

if $d_i < 0$,

$$M_i^{\text{binary}}(S) = \begin{cases} 0, & S < S_i, \\ 1, & S \geq S_i, \end{cases} \quad (16)$$

where

$$S_i = \frac{\beta_i - (lg(1/t_m - 1)/-A) - T_R}{d_i} \quad (17)$$

is the threshold on step size for flipping pixel i. At the current mask, if a pixel's value is less than t_m and its negative gradient is positive, or if a pixel's value is larger than t_m and its negative gradient is negative, then the pixel will be flipped when we apply a step size S larger than S_i. Other pixels are unflippable no matter how large step size S we use. So it is easy to determine how many pixels can be flipped. To control the number of pixels flipped during golden section search, we first mark all flippable pixels along the negative gradient direction. Then we calculate the threshold on step size, S_i, for each flippable pixel i. By sorting these thresholds from smallest to largest, the number of pixels flipped can be controlled by setting the value of S. For example, by using the 50th value of the sorted thresholds as the step size S, 50 pixels will be flipped along the negative gradient. In golden section search, the minimum and maximum sorted thresholds can be used to define the search region. In this paper, we use a segment in the region from the minimum to the maximum sorted thresholds as our search region. The details will be discussed in Section 5.

4.3. Jump Technique. Because of the noncyclic nature of our transformation, the solution space is not so rugged. But it is still extremely complicated with many local minima. As gradient-based search techniques will be trapped at a local minimum, we introduce a new technique named *jump* in order to jump out of the local minimum and continue the search. During the line search process, if the algorithm cannot find a better solution along the search direction (i.e., gets trapped at some local minimum), instead of terminating, it will jump along the search direction with a large step size to a probably worse solution. Then the algorithm will continue the gradient-based search starting from the new solution. If the step size is large enough, it is likely that the algorithm will not converge to the previous local minimum. At the end, the algorithm will return the best local minimum that has been found. For example, if 2 jumps happened, there would be 3 local minima, the first one was found without jump, and the other 2 were found by 2 jumps. Our program keeps recording the local minima and returns the best one at last.

4.4. Direct Rounding of Gray Mask. In order to apply gradient-based approach, it is unavoidable to relax the integer constraints. As a result, the optimized mask becomes a gray one. Because our goal is to generate a binary mask, the optimized gray mask has to be rounded to a binary one at last. A regularization framework was proposed in [14, Section IV.A] and also in [16, Chapter 6.1] to bias the output gray mask to be closer to binary. This regularization approach adds to the objective function (i.e., (6)) a quadratic penalty term for each pixel. However, it is still hard to control the change in the image pattern error caused by the rounding of the gray mask at the end. The optimized gray mask may achieve a low pattern error. However, after rounding the gray mask into binary, the pattern error often increases dramatically. Instead of using the quadratic penalty regularization framework, we propose to directly round the optimized gray mask into a binary one in each iteration before evaluating the pattern error. In this way, we simplify the objective function and also guarantee that our search will not be misled by inaccurate pattern error values. we observed that it works well based on our experiments.

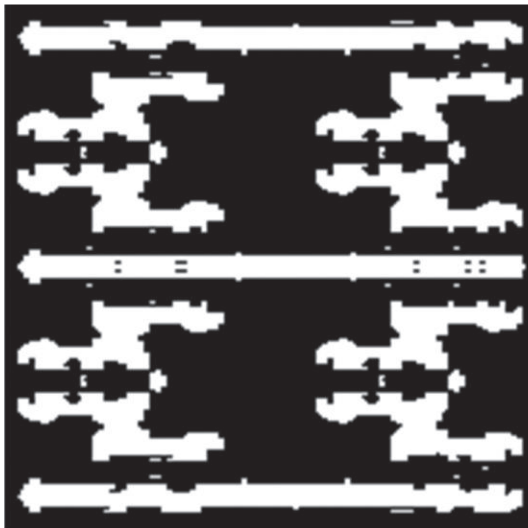
5. Experimental Results

We compare an implementation of our algorithm with the program developed by Ma and Arce [16]. Both of the programs are coded in Matlab and executed on an Intel Xeon(R) X5650 2.67 GHz CPU. The Matlab program of Ma and Arce [16] is public, and we downloaded it from the publisher. The runtime reported is CPU time, and the programs are restricted to use a single core when running in Matlab.

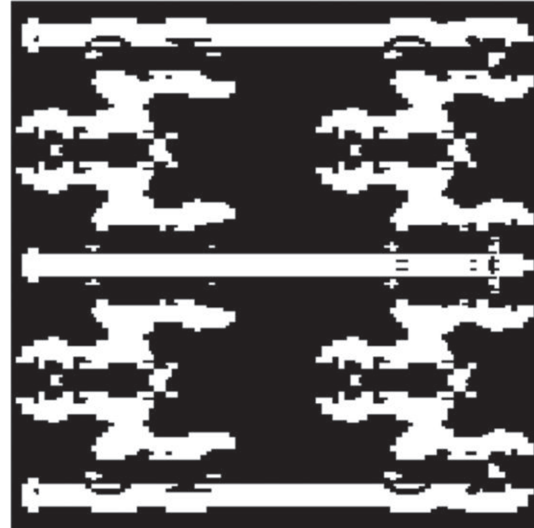
In [16], the program uses cosine transformation and a preset step size of 2. Besides, it applies regularization with a quadratic penalty term as mentioned in Section 4.4. Isolated perturbations, protrusions, and so forth are very hard to be written by the mask writer, so in [16], it also applies another regularization called complexity penalty term, which restricts the complexity of optimized binary mask. The details can be found in [14, Section IV.B] and also in [16, Chapter 6.2]. To have a fair comparison, we followed previous works and our program applies the complexity penalty regularization too. But as we mentioned in Section 4.4, our program does not apply the quadratic penalty regularization but directly rounds the optimized gray mask into a binary one instead whenever pattern error is evaluated. In [16], the stopping criteria of gradient-based search is set in an ad hoc manner according to the target pattern. In order to fairly compare the two programs on various masks, the same stopping criteria are applied to both programs. The criteria are that if the average pattern error over the last 30 iterations is larger than the average pattern error over the 30 iterations before that,

Table 1: Pattern and runtime comparison between [16] and ours.

No.	Mask size (pixel)	Pixel size (nm)	Feature size (nm)	Pattern error					Runtime (s)		
				[16]	(%)	[16] with our runtime	(%)	Ours	[16]	(%)	Ours
1	184 × 184	5.625	45	1512	(9.88)	1566	(13.81)	1376	192	(−49.61)	381
2	400 × 400	5.625	45	3308	(24.17)	3780	(41.89)	2664	13337	(13.24)	11778
3	2000 × 2000	5.625	45	108144	(8.55)	109267	(9.68)	99624	17918	(−0.38)	17986
4	2000 × 2000	5.625	45	61350	(10.72)	66516	(20.05)	55409	12457	(19.81)	10397
5	4000 × 4000	5.625	45	58410	(198.39)	58410	(198.39)	19575	78652	(16.34)	67603
6	2000 × 2000	4	32	101785	(75.18)	102100	(75.72)	58104	49333	(13.45)	43485
7	2000 × 2000	4	32	64252	(39.49)	75300	(63.47)	46063	51856	(−21.57)	66117
8	4000 × 4000	4	32	148356	(358.80)	148420	(358.99)	32336	62008	(43.35)	43255
9	4000 × 4000	4	32	52160	(153.30)	52160	(153.30)	20592	71489	(5.77)	67588
	Average				97.61		103.92			4.49	



(a) Optimized binary mask of [16]



(b) Our optimized binary mask

(c) Image pattern of optimized binary mask of [16], error = 1512

(d) Image pattern of our optimized binary mask, error = 1376

Figure 8: Comparison of optimized binary mask and pattern error for target pattern no. 1.

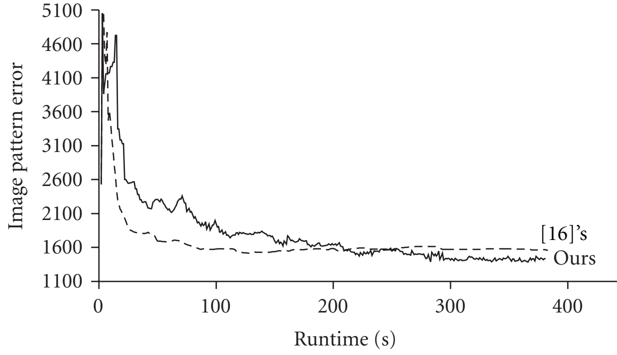


Figure 9: The convergence curve for target pattern no. 1.

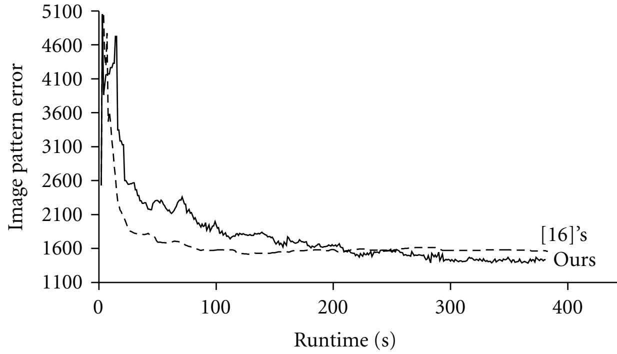


Figure 10: The convergence curve for target pattern no. 6.

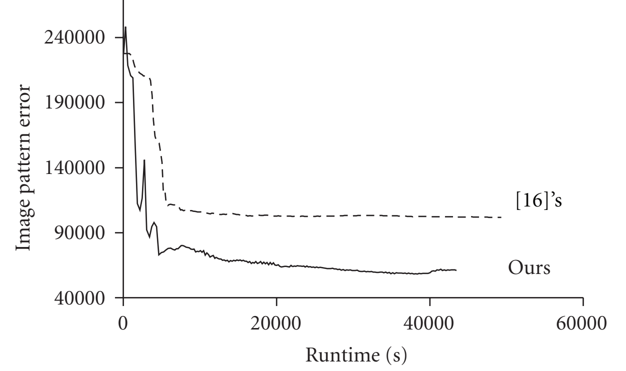


Figure 11: The convergence curve for target pattern no. 7.

the program will stop. For evaluation of pattern error in both programs, in each iteration, the optimized gray mask is rounded using (13). We use the same convolution kernel h as the Matlab program of [16].

For the photoresist model, we use $a = 25$ and $t_r = 0.19$ for the sigmoid function in (5). For the transformation of mask variables from β, we use $A = 4$ and $T_R = 0$ for the sigmoid function in (11). The threshold t_m in (13) is set to 0.5.

Based on our observation of many experiments, for the first iteration of gradient-based search, the minimum pattern error can almost always be achieved by flipping less than 10% of all pixels. One example is showed in Figures 6 and 7, where the minimum pattern error is at about 7.7%. Then in the later iterations, this region remains nearly the same or keeps shrinking. So for the first 2 iterations, we set the initial search region of golden section search to be the region in which the first 10% of overall pixels can be flipped along the negative gradient direction. Our program keeps recording the minimum location which is found in each iteration to guide the search region for the next iteration. For example, if in the current iteration, the minimum error is found at 5% of the overall pixels flipped, to be on the safe side, the search region of the next iteration will be automatically set as 1.5 times of 5%, which is 7.5%, of the pixels flipped. To prevent this search region from shrinking too small, we set a minimum as 2%. For the stop criteria of the golden section search, we set it as 0.25%, which means that when the search region shrinks to or below 0.25%, our program will stop searching. As mentioned above for the jump technique, if our program cannot find a better solution along the search direction until it stops searching (i.e., gets trapped at some local minimum), our program will take the best solution, except the starting point of that line search, as a new solution, although it is a worse solution. This means one jump.

The comparisons of pattern error of optimized binary masks and runtime between our program and that of [16] are shown in Table 1. We use 9 binary image patterns as predefined targets. The outer and inner partial coherence factors for 184×184 target pattern are set to 0.4 and 0.3, respectively; for 400×400 target pattern, are set to 0.975 and 0.8, respectively; for all 2000×2000 target patterns, are set to 0.3 and 0.299, respectively; and for all 4000×4000 target patterns, are set to 0.2 and 0.1995, respectively.

The pattern errors reported are calculated according to the best binary mask generated for both programs. All gradient-based methods strongly depend on a starting solution. We followed the previous works and used the target as the starting point to search. The runtimes listed in the last two columns are based on the stopping criteria mentioned above. As the table shows, our program always generates better optimized binary mask which has significantly less pattern error. The pattern errors of [16] are higher than ours by 8.55% to 358.80%, with an average of 97.61%. Moreover, the programn of [16] uses 4.49% more runtime than our program on average.

We report the pattern error of the final binary mask generated for the program of [16] with our program's runtime in column 6 of the table. For target patterns no. 1, no. 3 and no. 7, the program of [16] stops earlier than ours. To see if the program of [16] will converge to better solutions if more runtime is allowed, we change the stopping criteria to let it run until the same runtime as that of our program. The result shows that the error gets worse in all 3 cases. If the pattern error of the best binary mask generated is reported instead, the result will be exactly the same as in column 5. It indicates that the program fails to get out of the local minima even with more time.

Target pattern no. 1 is obtained from [16]. We illustrate the optimized binary masks and the corresponding image

patterns for both programs in Figure 8. The pattern error convergence curves are shown in Figure 9.

More pattern error convergence curves are shown in Figures 10 and 11 for target patterns no. 6 and no. 7, respectively.

Because the program we obtained from [16] is fine-tuned for the target pattern no. 1 which is also obtained from [16], the experiment result of our algorithm is not so much better than that of [16]. However, based on the experiment results of other target patterns which cover multiple mask sizes, pixel sizes, and feature sizes, our algorithm has an overwhelming advantage due to the application of line search engine which is enabled by our novel transformation of mask pixel. Based on the observation of Figures 10 and 11, the program of [16] is very easy to be trapped because line search is not applied and a fixed step size is used. On the other hand, benefited from line search and jump technique, our program has better performance. Even if our program is trapped, the jump technique enables the algorithm to jump out and continue the search.

6. Conclusion

In this paper, we introduced a highly efficient gradient-based search technique to solve the inverse lithography problem. We proposed a new noncyclic transformation of mask variables to replace the well-known cyclic one. Our transformation is monotonic, and it enables a much better control in flipping pixels and the use of line search to minimize the pattern error. We introduced a new technique named jump in order to jump out of the local minimum and continue the search. We used direct rounding technique to simplify the optimization. The experimental results showed that our technique is significantly more effective than the state of the art. It produces better binary masks in a similar runtime. The four techniques we proposed should be applicable to other iterative gradient-based search approaches, like the conjugate gradient method. We plan to incorporate our techniques into other search methods in the future.

References

[1] J. Z. Yan and C. Chu, "DeFer: deferred decision making enabled fixed-outline floorplanner," in *Proceedings of the 45th Design Automation Conference (DAC '08)*, pp. 161–166, June 2008.

[2] J. Z. Yan and C. Chu, "DeFer: deferred decision making enabled fixed-outline floorplanning algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 3, pp. 367–381, 2010.

[3] J. Z. Yan and C. Chu, "Optimal slack-driven block shaping algorithm in fixed-outline floorplanning," in *Proceedings of the ACM International Symposium on Physical Design (ISPD '12)*, pp. 179–186, 2012.

[4] J. Z. Yan, C. Chu, and W. K. Mak, "SafeChoice: a novel clustering algorithm for wirelength-driven placement," in *Proceedings of the ACM International Symposium on Physical Design (ISPD '10)*, pp. 185–192, March 2010.

[5] J. Z. Yan, C. Chu, and W. K. Mak, "SafeChoice: a novel approach to hypergraph clustering for wirelength-driven placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 7, pp. 1020–1033, 2011.

[6] J. Z. Yan, N. Viswanathan, and C. Chu, "Handling complexities in modern large-scale mixed-size placement," in *Proceedings of the 46th ACM/IEEE Design Automation Conference (DAC '09)*, pp. 436–441, July 2009.

[7] A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, SPIE Press, 2001.

[8] Y. Granik, "Fast pixel-based mask optimization for inverse lithography," *Journal of Microlithography, Microfabrication and Microsystems*, vol. 5, no. 4, Article ID 043002, 2006.

[9] A. Poonawala and P. Milanfar, "A pixel-based regularization approach to inverse lithography," *Microelectronic Engineering*, vol. 84, no. 12, pp. 2837–2852, 2007.

[10] X. Ma and G. R. Arce, "Pixel-based OPC optimization based on conjugate gradients," *Optics Express*, vol. 19, no. 3, pp. 2165–2180, 2011.

[11] X. Ma and G. R. Arce, "Generalized inverse lithography methods for phase-shifting mask design," *Optics Express*, vol. 15, no. 23, pp. 15066–15079, 2007.

[12] X. Ma and G. R. Arce, "PSM design for inverse lithography with partially coherent illumination," *Optics Express*, vol. 16, no. 24, pp. 20126–20141, 2008.

[13] M. Born and E. Wolf, *Principles of Optics*, Cambridge University Press, New York, NY, USA, 1999.

[14] A. Poonawala and P. Milanfar, "Mask design for optical microlithography—an inverse imaging problem," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 774–788, 2007.

[15] A. Poonawala and P. Milanfar, "OPC and PSM design using inverse lithography: a non-linear optimization approach," in *Optical Microlithography XIX*, vol. 6514 of *Proceedings of SPIE*, pp. 1159–1172, February 2006.

[16] X. Ma and G. R. Arce, *Computational Lithography*, John Wiley & Sons, New York, NY, USA, 2010.

[17] P. Yu and D. Z. Pan, "TIP-OPC: a new topological invariant paradigm for pixel based optical proximity correction," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '07)*, pp. 847–853, November 2007.

[18] B. E. A. Saleh and M. Rabbani, "Simulation of partially coherent imagery in the space and frequency domains and by modal expansion," *Applied Optics*, vol. 21, no. 15, pp. 2770–2777, 1982.

[19] L. Pang, Y. Liu, and D. Abrams, "Inverse Lithography Technology (ILT), What is the impact to photomask industry?" in *Proceedings of the Photomask and Next-Generation Lithography Mask Technology XIII*, vol. 6283 of *Proceedings of SPIE*, April 2006.