Mask Design for Optical Microlithography—An Inverse Imaging Problem

Amyn Poonawala and Peyman Milanfar, *Senior Member, IEEE*

***Abstract*—In all imaging systems, the forward process introduces undesirable effects that cause the output signal to be a distorted version of the input. A typical example is of course the blur introduced by the aperture. When the input to such systems can be controlled, prewarping techniques can be employed which consist of systematically modifying the input such that it (at least approximately) cancels out (or compensates for) the process losses. In this paper, we focus on the optical proximity correction mask design problem for "optical microlithography," a process similar to photographic printing used for transferring binary circuit patterns onto silicon wafers. We consider the idealized case of an incoherent imaging system and solve an inverse problem which is an approximation of the real-world optical lithography problem. Our algorithm is based on pixel-based mask representation and uses a continuous function formulation. We also employ the regularization framework to control the tone and complexity of the synthesized masks. Finally, we discuss the extension of our framework to coherent and (the more practical) partially coherent imaging systems.**

***Index Terms*—Image synthesis, inverse lithography, inverse problems, mask design, optical microlithography, optical proximity correction (OPC), regularization, sigmoid.**

I. Introduction

In all imaging systems, the underlying physical process introduces undesirable distortions which causes the output signal to be a warped version of the input. To overcome this, when the input is controllable, prewarping techniques can be employed which consist of systematically modifying the input such that it will cancel out the process losses. Thus, in effect, we are precompensating for the process distortions to come. This is an *image synthesis* [1], [2] problem which consists of finding an image that when used as the input to a given imaging system, results in the desired output image (to within some prescribed tolerance).

The problems of *image restoration* (reconstruction) and *image synthesis* (design) are related but not the same. In both cases, the output and the imaging system are known but the input is unknown [1], [2]. In the restoration case, the output image results from an actual but unknown input image [3], and, therefore, at least one solution must exist in the absence of measurement error or noise. However, in the synthesis case, there may be no input capable of producing the prescribed output, and, hence, existence of a solution is a concern [1], [2]. Furthermore, if a solution does exist, it may not be unique.

The image formation process can be mathematically expressed as

$$z(x,y) = T\{i(x,y)\} \tag{1}$$

where $T\{.\}$ is the forward model which maps the input intensity function $i(x,y)$ to the output intensity function $z(x,y)$. Let $z^*(x,y)$ be the desired output intensity function. We seek to find a predistorted input intensity function which will give us an approximation to the desired output $z^*(x,y)$. This is achieved by searching the space of all inputs and choosing $\hat{i}(x,y)$ which minimizes a distance $d(z(x,y), z^*(x,y))$, where $d(\cdot,\cdot)$ is some appropriate distance metric to be defined later. To summarize, we write

$$\hat{i}(x,y) = \arg\min_{i(x,y)} d\left(z^*(x,y), T\{i(x,y)\}\right). \tag{2}$$

In this paper, we focus on studying prewarping techniques to design optical proximity correction (OPC) [4] masks for microlithography. We follow the inverse imaging mask design approach which consists of inverting the mathematical forward model from mask to wafer. Section II discusses the optical lithography procedure and introduces the forward model employed to simulate the distortion effects of the imaging system. The optimization procedure is discussed in Section III, and, in Section IV, we introduce two regularization functions to obtain two-toned and easy to manufacture masks, respectively. We discuss the weight mask and selected parameter optimization in Section V, and present conclusive remarks and future extensions in Section VI.

II. Optical Microlithography and OPC

A. Optical Microlithography

Optical microlithography, a process similar to photographic printing, is used for transferring circuit patterns onto silicon wafers and forms a very critical step in the IC fabrication flowchart [5]. As illustrated in Fig. 1, the pattern to be replicated on the wafer is first formed on a reticle (stencil or mask). An illuminator (UV source) is shone through this mask producing an image of the pattern through the lens system, which is eventually projected down onto a photoresist-coated silicon wafer using a projection system (typical reduction factor of 4X). The photoresist is chemically sensitive to light, and, hence, only the exposed regions are developed away leaving behind a likeness of the mask pattern on the substrate (wafer).

Manuscript received December 2, 2005; revised October 4, 2006. This work was supported by Intel. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Amir Said.

A. Poonawala is with the Computer Engineering Department, University of California, Santa Cruz, CA 95064 USA (e-mail: amyn@soe.ucsc.edu).

P. Milanfar is with the Electrical Engineering Department, University of California, Santa Cruz, CA 95064 USA (e-mail: milanfar@ee.ucsc.edu).

Digital Object Identifier 10.1109/TIP.2006.891332

1057-7149/$25.00 © 2007 IEEE

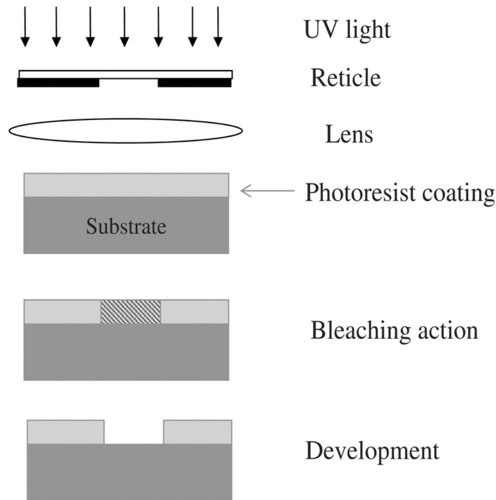

Fig. 1. Steps involved in optical microlithography.

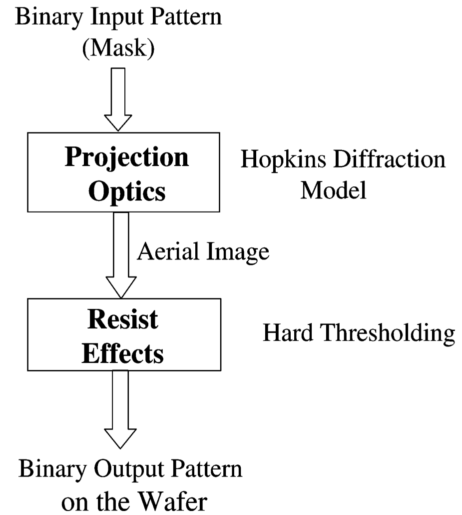

Fig. 2. Simplified view of the lithography process.

In an ideal world, the output circuit pattern on the wafer will be identical to the mask pattern. Unfortunately, the band-limited imaging system introduces distortions caused by the diffraction effects of the lens-projection system, particularly at higher resolutions. The optical projection system acts as a "low-pass" spatial frequency filter. These undesirable distortions lead to a loss of pattern fidelity, and, hence, the output pattern obtained on the wafer is a distorted (blurred) version of the input [4] which can lead to short circuits, defects, and yield loss. The semiconductor industry strives to adhere to the International Technology Roadmap for Semiconductors (ITRS) [6] which has been driven by Moore's law for the past three decades. Moore's law with regard to lithography requires the critical dimension (CD)[1] to shrink by 30% every two years (currently CD $= 65$ nm). This puts very stringent requirements on lithography thereby making it one of the most challenging problems faced by the semiconductor industry.

The resolution (*Res*) of the lithography system in Fig. 1 can be expressed using Rayleigh's criterion as follows:

$$Res = \frac{k_1 \lambda}{NA}. \tag{3}$$

The numerical aperture $NA = n \sin(\theta)$ where n is the refractive index of the medium, and θ is the half angle of the maximum cone of light that can enter the lens. Obviously, the resolution can be improved by increasing the numerical aperture of the imaging system (NA) or decreasing the wavelength (λ) and these alternatives have been actively explored by researchers in the lithography community. Immersion lithography is a very recent advancement [7] which increases NA by using water or other transparent fluid with higher refractive index. The current values for λ and NA are 193 nm and around 0.9 (dry lithography), respectively. Pushing the above parameters values beyond this limit is very costly, risky, and time consuming. Therefore, it has been established that the most important, feasible, and promising method to achieve the goal of enhanced resolution is by decreasing the process constant k_1 using resolution enhancement techniques (RETs) [4], [8], [9].

[1]Critical dimension refers to the smallest dimension on the layout, typically the microprocessor transistor gate length or the half-pitch of DRAM.

The commonly employed RETs include off-axis illumination, phase-shift methods [10], and double-exposure techniques [11]. Any type of RET, while allowing for an increased resolution, introduces image distortions dependent on the geometries within a certain proximity radius and the optical conditions. OPC is a very commonly employed technique for compensating the distortions caused by RETs in optical lithography. It is important to note that phase-shift methods and OPC are implemented by making physical changes to the mask (reticle). Unfortunately, these changes increase the complexity, data-size and cost of the masks (currently in several millions for a complete mask set having approximately 32 levels), and, hence, the challenge is to implement RETs while taking the above factors into account.

B. *Optical Proximity Correction*

In this paper, we focus on designing OPC masks to overcome the distortions occurring in optical microlithography. OPC consists of adding subresolution features to the original layout. Thus, we precompensate for the process losses by modifying the original layout, which leads to better pattern fidelity and improved resolution (see Fig. 6) [4], [12].

OPC has been carried out mainly using two approaches; rule-based, or model based. These are often used in conjunction depending on the speed, accuracy, throughput, cost, and manufacturability requirements. As the name suggests, in the rule-based OPC scheme, empirical rules are developed to counteract the commonly occurring problems around pattern corners, edges, local interactions, etc. [13]. These are then applied throughout the pattern to provide a general improvement in pattern fidelity. Rule-based OPC techniques are simple to implement, and have relatively low-cost and high throughput. They are very popular in the semiconductor industry and are widely employed in several technology nodes. However, they only compensate for local features and do not optimize global performance depending on the overall layout.

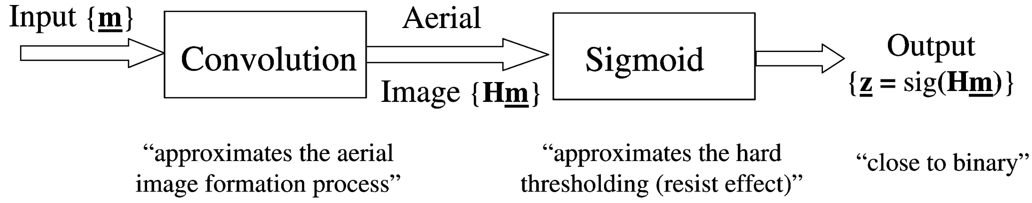

Fig. 3. Approximated forward process model.

Model-based methods, on the other hand, use a mathematical description to represent the warping process (forward model). As such, they are more universal and represent a more aggressive OPC strategy which will be required for 45- and 32-nm nodes [14]. The success of these methods relies heavily on accurate modelling of the distortion process which has been studied carefully by physicists, chemists, and other researchers in the lithography community [15]. Fig. 2 illustrates a simplified view of the lithography process. It consists of two functional blocks; namely, the projection optics effects (aerial image formation), and the resist effects. The former is simulated using the Hopkins scalar (or vector) model for partially coherent imaging systems [16], [17]. The resist effects are simulated using Dill's model [5], Mack's Model [18], [19], the constant threshold resist (CTR) model [20], the variable threshold resist (VTR) model [21], or other models as in [22]. We use the simplest (CTR) model in our analysis.

Model-based OPC itself has two flavors. The forward model-based OPC techniques suggested by Cobb and Zakhor [23] parameterize the mask pattern using edges and corners and proceed by nudging these geometric elements while simulating the output wafer pattern (using the forward model) until certain criteria are satisfied. Backward (or inverse) model-based techniques (which form the topic of interest of this article) invert the mathematical model and attempt to directly synthesize the optimized mask pattern [10]. In the past, Sherif, *et al.* [24], Liu-Zakhor [25], and Pati-Kailath [26] used branch and bounds methods, simulated annealing, and projection on convex sets (POCS), respectively, to synthesize the mask. More recently, Oh *et al.* [27] used random-pixel flipping and Erdmann *et al.* [28] proposed genetic algorithms to solve the above problem. Granik [29] recently used nonlinear programming to solve the above problem.

In our approach, initially reported briefly in [30], and significantly expanded here, we depart from our predecessors by formulating the mask synthesis problem using continuous function optimization and using the gradient information to systematically search the solution space. We use the pixel-based parametrization which results in more flexibility in mask representation. However, this approach also suffers an inherent drawback in that the synthesized masks are very difficult to manufacture and inspect. Therefore, we employ the regularization framework and use an L_1 norm-based penalty function to curb the mask complexity. Furthermore, we also employ a quadratic penalty term to ensure that the synthesized mask is restricted to be binary. The gradient of the cost function can be calculated analytically, and has a computational complexity $O(N^2 \log(N))$ for N^2 pixels.

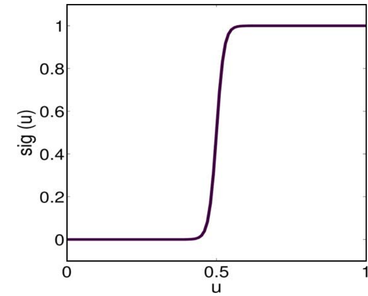

Fig. 4. Sigmoid function $\mathrm{sig}(u) = 1/(1 + e^{-80(u-0.5)})$.

C. Proposed Method for Mask Synthesis

In this paper, we treat the problem as an "inverse" problem, and focus on developing fast and efficient methods for backward (or inverse) model-based OPC. We use a pixel-based mask representation; hence the input, output, and desired patterns are all represented using discrete 2-D images. We employ the approximated forward process model illustrated in Fig. 3 in our analysis.[2] Note that the aerial image formation step from Fig. 2 is approximated using the convolution of the input pattern with a 2-D Gaussian kernel for the sake of simplicity. Second, we employ the sigmoid type transfer function instead of the hard-thresholding (heaviside) operation to simulate the resist effect. Using a Heaviside operator (hard threshold) defined as

$$\Gamma(u) = \begin{cases} 0, & u \leq t_r \\ 1, & u > t_r \end{cases} \tag{4}$$

would result in a discrete combinatorial optimization problem. A sigmoid, however, is a smooth, continuous function which can arbitrarily closely approximate the Heaviside function [31]. With this choice, we can use gradient-based continuous function optimization techniques like steepest-descent to solve the mask design problem.

In particular, we employ the logarithmic sigmoid function

$$\mathrm{sig}(u) = \frac{1}{1 + e^{-a(u-t_r)}} \tag{5}$$

where the parameter a dictates the steepness of the sigmoid. A large value of a leads to a very steep sigmoid which closely resembles the hard thresholding operation. The parameter t_r is the threshold parameter of the sigmoid and is set equal to the threshold level of the resist in accordance with the CTR model. As an example, Fig. 4 illustrates the behavior of a sigmoid with $a = 80$ and $t_r = 0.5$.

[2] $\underline{\mathbf{m}}$ in Fig. 3 denotes the mask which plays the part of the input intensity function $i(x, y)$ used in (1) and (2).

D. Limitations and Goals of This Work

Before we delve into the details of our algorithm, we would like to discuss some goals and shortcomings of our work. Sayegh *et al.* [1], [2] formulated the *image construction* (or the inverse lithography) problem in the early 1980s and saw parallels with the well-known *image restoration* problem. Our objective here is to demonstrate how some of the well studied tools for image restoration like regularization and nonlinear optimization can be employed to solve the inverse lithography problem. Most of the past work on pixel-based mask design employed techniques like numerical differentiation, integer programming [24], [32], simulated annealing [25], POCS [33], and random-pixel flipping [27]. Our goal is to formulate mask design as a continuous function optimization problem and analytically calculate the gradient. The gradient (when calculated analytically) has a computational complexity of only $O(N^2 \log(N))$ making this an attractive approach.

The current work takes the idealized view of the lithography system consisting of a band-limited linear shift-invariant (incoherent imaging) system followed by a hard-limiter. We employ a very generic Gaussian convolution model to calculate the aerial image. Note that this does not capture the physical effects in real-world microlithography systems which are partially coherent in nature. Aerial image calculations for partially coherent imaging systems employ the Hopkins model [16] which is nonlinear. The CTR model is a simplified way of matching the aerial image contours. Typical lithography simulators model the resist effect using the VTR [21] or empirical resist models which are analytically noninvertible. Hence, CTR model is commonly employed for contour matching while solving the inverse lithography problem [2], [24], [29], [34].

Our algorithm in its present form cannot be directly employed to solve the practical real-world lithography problem and fulfilling all the rigorous industry requirements. However, our goal is to solve an inverse imaging problem which is a representative of the optical lithography problem. We want to demonstrate the potential and applicability of the vast pool of inverse imaging techniques and nonlinear programming to solve the mask design problem. Finally, we would like to highlight that our proposed framework is not limited to the Gaussian model. In Section VI-A, we discuss the extension of our framework to the more general cases of coherent and partially coherent imaging systems.

III. Model-Based Optimization

We define vectors $\underline{\mathbf{z}}^*, \underline{\mathbf{z}}, \underline{\mathbf{m}} \in \Re^{N^2\times 1}$ which are obtained by sampling and lexicographic ordering of $z^*(x,y)$, $z(x,y)$, and $m(x,y)$ respectively. Throughout our discussion, $\underline{\mathbf{z}}^*$ represents the prescribed (desired) *binary* pattern, $\underline{\mathbf{z}}$ represents the *gray-level* output pattern, and $\underline{\mathbf{m}}$ represents the input pattern fed to the imaging system (can be *binary* or *gray-level*). The forward model in Fig. 3 can be mathematically represented as

$$\underline{\mathbf{z}} = \mathrm{sig}(\mathbf{H}\underline{\mathbf{m}}) \tag{6}$$

where $\mathbf{H} \in \Re^{N^2\times N^2}$ is the blur matrix representing the point spread function (PSF) of the imaging system, $\underline{\mathbf{z}} = [z_1,\ldots,z_{N^2}]^T$, and $\underline{\mathbf{m}} = [m_1,\ldots,m_{N^2}]^T$. Thus, every pixel undergoes a cascade of convolution followed by the sigmoidal transformation. To write (6) another way

$$z_i = \frac{1}{1+\exp\left[-a\left(\sum_{j=1}^{N^2} h_{ij}m_j\right)+at_r\right]} \tag{7}$$

for $i = 1,\ldots,N^2$.

A. Optimization Problem

To begin, we formulate the OPC-mask design problem as finding the optimized mask layout $\hat{\underline{\mathbf{m}}}$ that minimizes the cost function $F(\underline{\mathbf{m}})$, defined as the L_2 norm of the difference between the desired pattern $\underline{\mathbf{z}}^*$ and the output pattern $\underline{\mathbf{z}}$. That is

$$\begin{aligned}\hat{\underline{\mathbf{m}}} &= \arg\min_{\underline{\mathbf{m}}} \{F(\underline{\mathbf{m}})\} = \arg\min_{\underline{\mathbf{m}}} \|\underline{\mathbf{z}}^* - \underline{\mathbf{z}}\|_2^2 \\ &= \arg\min_{\underline{\mathbf{m}}} \|\underline{\mathbf{z}}^* - \mathrm{sig}(\mathbf{H}\underline{\mathbf{m}})\|_2^2.\end{aligned} \tag{8}$$

Later, in Section IV, we refine this approach by introducing the regularization terms and augmenting the cost function. We can write (8) as

$$\hat{\underline{\mathbf{m}}} = \arg\min_{\underline{\mathbf{m}}} \sum_{i=1}^{N^2} (z_i^* - z_i)^2 \tag{9}$$

where z_i is defined using (7).[3]

Note that $\underline{\mathbf{m}}$ consists of the transmission values of a binary mask which can only take values 0 or 1, resulting in a combinatorial optimization problem. However, to make the problem analytically tractable, for the time being, we relax the parameter values to lie in the *range* [0,1]. This is achieved by imposing the following inequality constraints on the optimization problem given in (8):

$$0 \le m_j \le 1 \quad \text{for } j = 1,\ldots,N^2. \tag{10}$$

The bound-constrained optimization problem can be further reduced to an unconstrained optimization problem using the following parametric transformation

$$m_j = \frac{1+\cos(\theta_j)}{2} \quad \text{for } j = 1,\ldots,N^2 \tag{11}$$

where $\underline{\boldsymbol{\theta}} = [\theta_1,\ldots,\theta_{N^2}]^T$ is the unconstrained parameter vector. The re-parameterized cost function can be formulated in terms of the parameter vector $\underline{\boldsymbol{\theta}}$ as follows:

$$F_1(\underline{\boldsymbol{\theta}}) = \sum_{i=1}^{N^2} \left(z_i^* - \frac{1}{1+\exp\left[-a\left(\sum_{j=1}^{N^2} h_{ij}\frac{1+\cos(\theta_j)}{2}\right)+at_r\right]}\right)^2. \tag{12}$$

We now employ the steepest-descent method to find the solution of the above problem which involves finding the first order derivatives of (12). As shown in the Appendix, the gradient

[3]As an alternative, it is also possible to consider the weighted least squares approach where weights are chosen according to the importance of different parts of the pattern. This may be viewed as a mix of model-based and rule-based approaches, where the rules determine weights in the model. We discuss this approach in Section V.

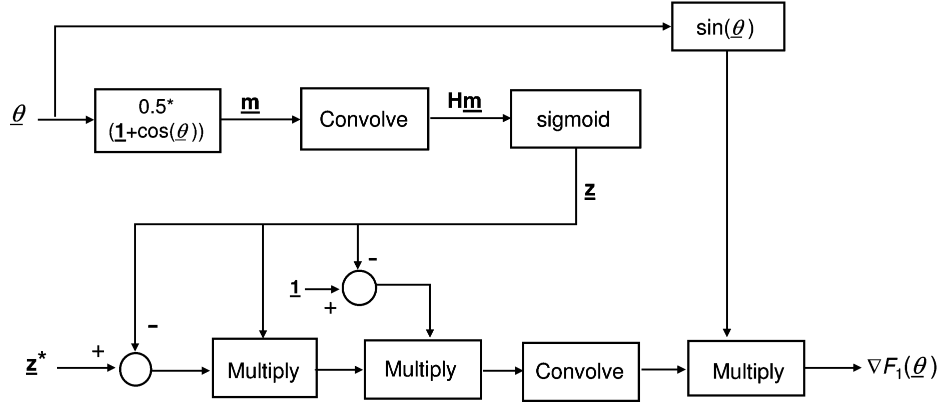

Fig. 5. Flowchart showing the sequence of operations required for calculating the gradient $\nabla F_1(\underline{\boldsymbol{\theta}})$. Note that *multiply* indicates element-by-element multiplication of 2-D arrays.

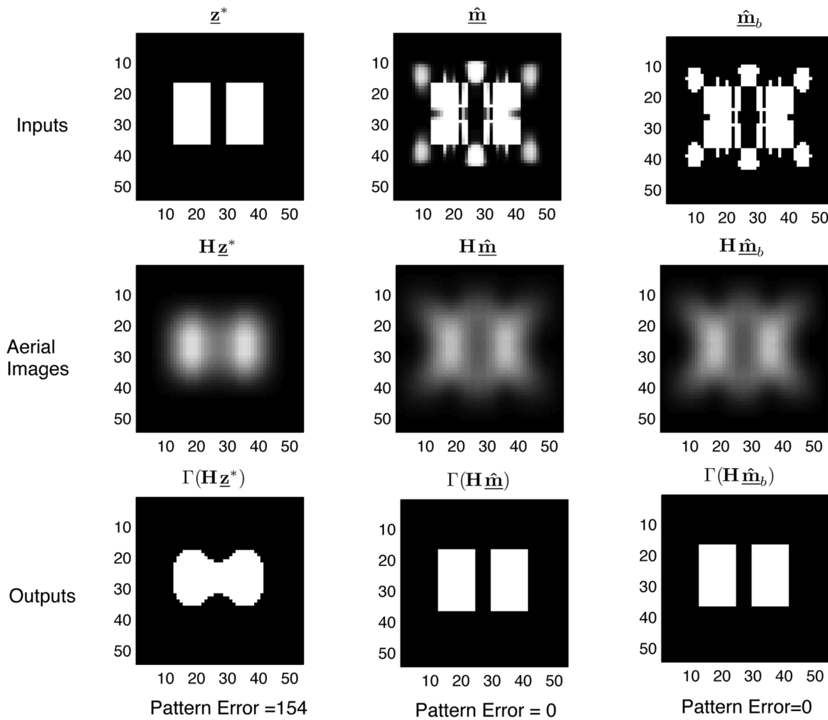

Fig. 6. Top row (input masks), left to right: Original pattern, optimized gray pattern, and binary pattern obtained using an optimum global threshold t_m. The middle and bottom rows indicate the corresponding aerial images and binary output patterns. The parameters are $a = 90$, $t_r = 0.5$, 15 × 15 filter with $\sigma = 5$ pixels, $s = 0.4$, and number of iterations $= 200$.

vector $\nabla F_1(\underline{\boldsymbol{\theta}}) \in \Re^{N^2 \times 1}$ can be calculated using the following expression:

$$\nabla F_1(\underline{\boldsymbol{\theta}}) = \underline{\mathbf{d}} = a\left(\mathbf{H}^{\mathbf{T}}\left[(\underline{\mathbf{z}}^* - \underline{\mathbf{z}}) \odot \underline{\mathbf{z}} \odot (\mathbf{1} - \underline{\mathbf{z}})\right]\right) \odot \sin(\underline{\boldsymbol{\theta}}) \tag{13}$$

where $\odot$ is the element-by-element multiplication operator. Note that the computational complexity of the gradient calculation is governed by the convolution operation. Therefore, the algorithmic complexity is $O(N^2 \log(N))$ where N^2 is the total number pixels in the image.

The kth iteration of steepest descent is given as

$$\underline{\boldsymbol{\theta}}^{k+1} = \underline{\boldsymbol{\theta}}^k - s\underline{\mathbf{d}}^k \tag{14}$$

where s is the step-size. The flowchart in Fig. 5 illustrates the steps involved in calculating the gradient given by (13). We would like to highlight the useful fact that due to the structure of (13), the steepest descent iterations can be quickly and directly carried out on the 2-D image array (matrices) with no need for the (algebraically convenient) raster scanning operation (see Fig. 5). More importantly, since we know the solution to be a perturbation of the prescribed pattern, the iteration can be initialized with the prescribed pattern, leading to quick convergence. The optimized pattern $\underline{\widehat{\mathbf{m}}}$ can finally be obtained from $\underline{\boldsymbol{\theta}}$ using (11).

We note that the pattern $\underline{\widehat{\mathbf{m}}}$ obtained using the above method is not binary. Instead each pixel can have gray values anywhere in [0,1]. This makes the resulting mask practically unrealizable, and, hence, we need a postprocessing step to obtain the synthesized *binary* OPC mask $\underline{\widehat{\mathbf{m}}}_b$. The simplest way to obtain $\underline{\widehat{\mathbf{m}}}_b$ from $\underline{\widehat{\mathbf{m}}}$ is using a global threshold parameter t_m, such that the

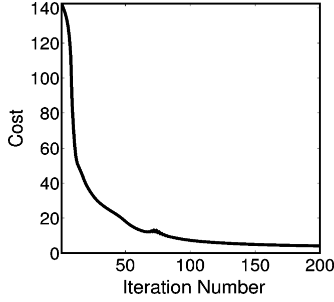

Fig. 7. Cost function versus steepest-descent iterations for Fig. 6.

error between $\underline{\mathbf{z}}^*$ and the output binary pattern obtained using $\widehat{\underline{\mathbf{m}}}_b$ (as the input) is minimized. The optimum value of t_m can be obtained using a simple line search operation [35]. However, the above approach for binarization is suboptimal and, in Section IV, we propose an alternative method for this step.

B. Some Results

We now present some simulations for the OPC masks synthesized using the aforementioned method.

The bottom row in Fig. 6 illustrates the binary output patterns (after resist thresholding) obtained using (left) the original pattern (i.e., $\underline{\mathbf{m}} = \underline{\mathbf{z}}^*$), (center) the optimized gray pattern ($\underline{\mathbf{m}} = \widehat{\underline{\mathbf{m}}}$), and (right) the binary pattern ($\underline{\mathbf{m}} = \widehat{\underline{\mathbf{m}}}_b$) as inputs. The corresponding aerial images (which form an intermediate step in the process flow) are illustrated in the middle row. The PSF effect was simulated using a Gaussian filter of size 15 $\times$ 15 with $\sigma = 5$ pixels, with the sigmoid parameters $a = 90$, $t_r = 0.5$ and the gradient descent iterations were carried out with step size $s = 0.4$. The final output binary patterns were obtained by thresholding the aerial images using the heaviside operator given in (4) with $t_r = 0.5$. Fig. 7 illustrates the cost function behavior for the first 200 iterations of steepest descent, indicating quick convergence. The algorithm was coded using Matlab and the execution time was 7 s on a 1.4-GHz Pentium-M processor.

The cost $F(\underline{\mathbf{m}})$ associated with the above three input patterns can be calculated using (8), where, for this example, $F(\underline{\mathbf{z}}^*) = 142.76$, $F(\widehat{\underline{\mathbf{m}}}) = 4.09$, and $F(\widehat{\underline{\mathbf{m}}}_b) = 8.46$. However, for the given imaging system (see Fig. 2), the aerial image is actually subjected to the Heaviside operation to obtain the binary output image. Hence, it is more appropriate to express the error using the binary output image instead of the sigmoidal transformed (gray level) output image. We employ a metric we call *pattern error*, defined as the total number of pixels which are not faithfully reproduced in the binary output pattern[4]

$$\text{error}(\underline{\mathbf{m}}) = \|\underline{\mathbf{z}}^* - \Gamma(\mathbf{H}\underline{\mathbf{m}})\|_1 \tag{15}$$

where $\Gamma(\cdot)$ is the heaviside operator defined in (4). From Fig. 6, we observe that $\text{error}(\underline{\mathbf{m}}) = 154$, $\text{error}() = 0$, and $\text{error}(\widehat{\underline{\mathbf{m}}}_b) = 0$.

Note that the imaging system was incapable of rendering the two bars distinguishable if the desired image is itself used as the input mask (i.e., $\underline{\mathbf{m}} = \underline{\mathbf{z}}^*$). However, the patterns are reproduced very faithfully using the synthesized OPC mask. The results also indicate a perfect corner reproduction which is physically unrealizable. The perfect resist reproduction appears because of the coarse sampling of the underlying pattern coupled with the hard thresholding operation. The current pixel-size is too big to capture the rounding at the corners. However, if we choose a smaller sampling interval, the corners wont be *perfectly* reproduced.

Fig. 8 illustrates the input and output patterns obtained by directly solving the bound-constrained optimization problem in (8) instead of performing the parametric transformation. We employed the *gradient projection algorithm* described in [36] and utilized the code provided by the author for this experiment. The experimental parameters are exactly the same as Fig. 6 and 200 iterations were performed. We observe that the optimized mask $\widehat{\underline{\mathbf{m}}}$ is structurally very similar albeit much more gray compared to one obtained in Fig. 6. As a result, the *pattern error* increases from 4 to 40 after the binarization step rendering it ineffective. A similar behavior was observed even for other test patterns. The above comparison illustrates that the parametric transformation technique also has an added indirect advantage; the unconstrained optimization routine tends to produce near-binary results, thereby easing the postprocessing binarization step. We attribute this to the cosine term in (11), which results in a nonlinear mapping onto [0,1] while favoring values closer to the extremities. Note that the above example is only for illustration purposes and all the remaining simulations in this article are carried out using (13) and (14).

Fig. 9 illustrates the input and output patterns for a more complicated circuit pattern. The Gaussian filter employed has size 11 $\times$ 11 with $\sigma = 14$ pixels (the high value of σ gives the effect of a severe averaging filter), while the sigmoid parameters are $a = 80$, $t_r = 0.5$, and $s = 0.5$. We observe that our algorithm performs an excellent job in prewarping and the output is extremely faithful to the desired pattern (*pattern error* 447 versus 1 pixel). Note that the binarization step employed from Section III-A increases the error from 1 to 18. Fig. 10 illustrates the cost function behavior for the first 200 iterations of the steepest-descent procedure. Note that the long vertical bar (on the extreme left) in the prescribed pattern in Fig. 9 is completely missing when reproduced using $\underline{\mathbf{m}} = \underline{\mathbf{z}}^*$. However, our optimization procedure starts adding prewarping elements to correct it at around 85th iteration, which results in a steep decrease in the cost function observed in Fig. 10. The kinks in the curve are owing to the large step size and can be overcome by choosing a smaller s.

The imaging system given in Fig. 2 is a binary-in-binary-out (BIBO) system. The binary output part is modelled by using the sigmoid-based function (see Fig. 3) which guarantees that the output pattern is always close to binary. The current setup does not incorporate the fact that the estimated input pattern should also be binary. The disadvantage is the need for an extra postprocessing step (binarization) which is suboptimal with no guarantee that the *pattern error* will be under control (see Fig. 9). There is also a possibility to jointly optimize the binarizing threshold and the gray pattern. In the next section, we use the regularization framework to overcome these problems and complete the binary-to-binary loop.

[4]Although we define (15) using L_1 norm, note that for binary patterns, L_0, L_1, and L_2 norms are all the same.

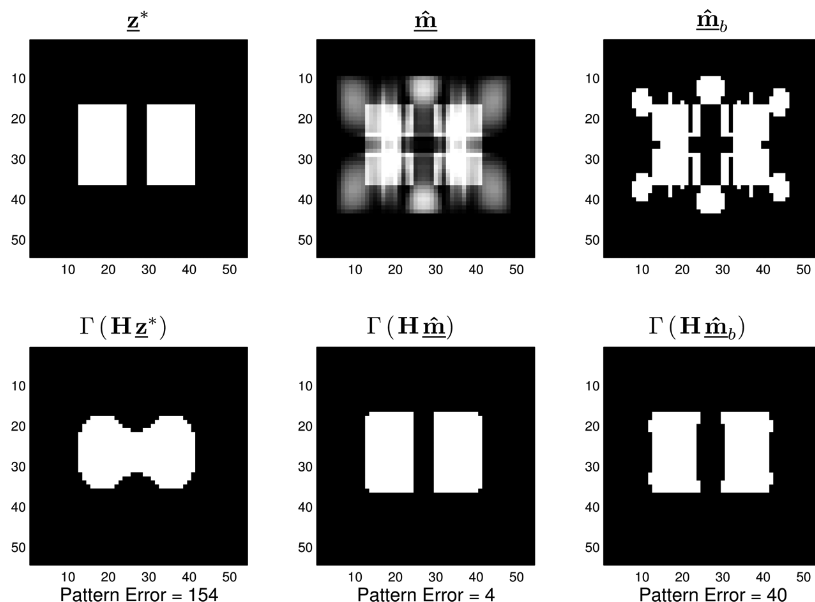

Fig. 8. Top row, left to right: Original pattern, optimized gray pattern using the constrained optimization routine from [36], and the binary pattern obtained using an optimum global threshold t_m. The bottom row indicates the corresponding binary output patterns. The parameters are $a = 90, t_r = 0.5$, 15 × 15 filter with $\sigma = 5$ pixels.

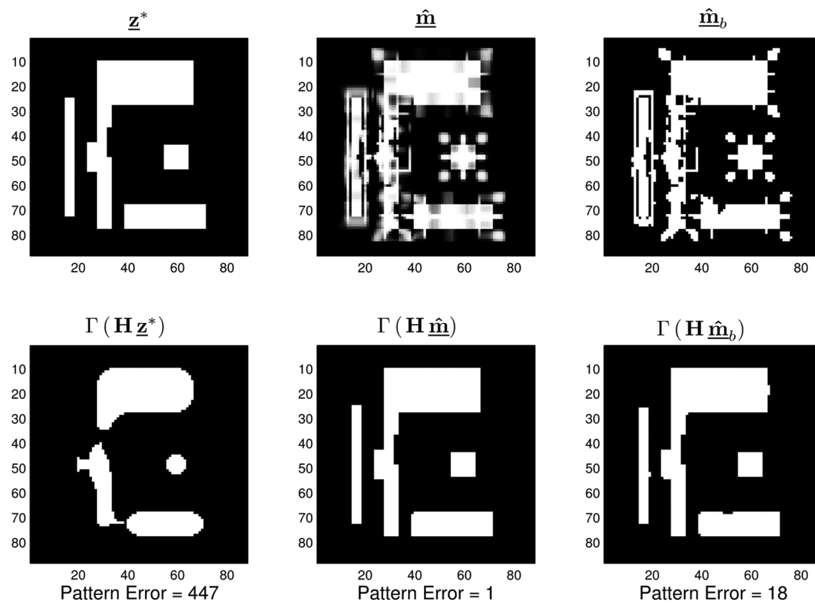

Fig. 9. Top row, left to right: Original pattern, optimized gray pattern, and binary pattern obtained using an optimum global threshold t_m. The bottom row indicates the corresponding binary output patterns. The parameters are $a = 80, t_r = 0.5$, 11 × 11 filter with $\sigma = 14$ pixels, $s = 0.5$, and number of iterations $= 200$.

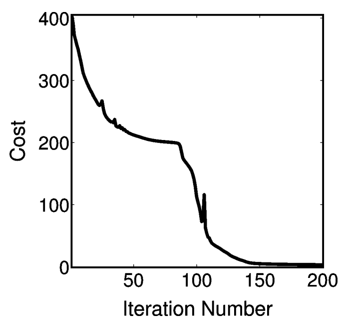

Fig. 10. Cost function versus steepest-descent iterations for Fig. 9.

IV. Regularization

If we look back again at Fig. 6, we observe that both the gray-level and binary input patterns give rise to the same (desired) binary pattern at the output. The BIBO mask design problem can itself have multiple solutions and the continuous domain formulation implies that now there can be infinitely many different input patterns, all giving rise to the same binary pattern at the output. However, we want our estimated pattern to satisfy certain properties which can be incorporated as prior knowledge about the desired solution. These properties can constrain the space of solutions to obtain a general desirable solution using

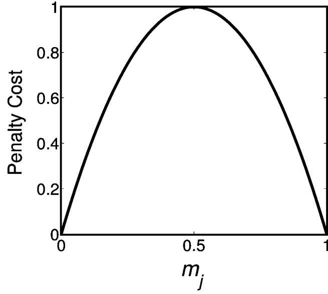

Fig. 11. Quadratic penalty cost function $R_{\rm bin}(\underline{\mathbf{m}})$.

an appropriate regularization term [37]. In general, the formulation can be described as follows:

$$\widehat{\underline{\mathbf{m}}} = \arg\min_{\underline{\mathbf{m}}} \left[F(\underline{\mathbf{m}}) + \gamma R(\underline{\mathbf{m}})\right] \tag{16}$$

where $F(\underline{\mathbf{m}})$ is the data-fidelity term, $R(\underline{\mathbf{m}})$ is the regularization function used to direct the unknown parameter $\underline{\mathbf{m}}$ towards the desired solution space, and γ is the user-defined scalar for adequately weighing the first (fidelity) term against the second (regularization) term.

In this section, we discuss two regularization terms corresponding to two desirable properties we chose to promote in our estimated solution. The first property discussed earlier is that our estimated mask should be (close to) binary, and the second is that our masks should be simple and, therefore, cheap and easy to manufacture.

A. Quadratic Penalty Term

The first penalty term employed to obtain near binary patterns is as follows:

$$R_{\rm bin}(\underline{\mathbf{m}}) = \sum_{j=1}^{N^2} \left[1 - (2m_j - 1)^2\right] = 4\underline{\mathbf{m}}^T(\underline{\mathbf{1}} - \underline{\mathbf{m}}) \tag{17}$$

where $\underline{\mathbf{1}} = [1, \ldots, 1]^T \in \Re^{N^2}$. Thus, every pixel m_j has an associated penalty given by the quadratic function (see Fig. 11)

$$r(m_j) = 1 - (2m_j - 1)^2.$$

As described earlier the mask transmission values will be constrained to lie in [0,1], and, hence, we are only interested in the behavior of the cost function in that range. The penalty incurred is zero for transmission values 0 or 1 and increases as we move away from binary region in either direction (maximum at $m_j = 0.5$).

The gradient $\nabla R_{\rm bin}(\underline{\mathbf{m}}) \in \Re^{N^2 \times 1}$ of the quadratic penalty term is given by

$$\nabla R_{\rm bin}(\underline{\mathbf{m}}) = (-8\underline{\mathbf{m}} + 4) \tag{18}$$

which can be used in conjunction with (13) and (14) while carrying out the steepest-descent iterations as before.

Fig. 12 illustrates the results with the same prescribed pattern, PSF and sigmoidal parameters used in Fig. 9 (Section III-B). However, here we employ the quadratic regularization term and minimize the overall cost function

$$J(\underline{\mathbf{m}}) = F(\underline{\mathbf{m}}) + \gamma_{\rm bin} R_{\rm bin}(\underline{\mathbf{m}})$$

where $F(\underline{\mathbf{m}})$ is defined in (8) and $R_{\rm bin}(\underline{\mathbf{m}})$ is defined in (17), with $\gamma_{\rm bin} = 0.015$. The masks in Fig. 12 were obtained using 200 iterations with $s = 1.5$. Note that unlike Fig. 9, the optimized gray-pattern is very close to binary. Hence, there is no need for the line search operation discussed in Section III-A to find the optimal global threshold parameter t_m. In fact, we can simply obtain the synthesized binary pattern $\widehat{\underline{\mathbf{m}}}_b$ by thresholding $\widehat{\underline{\mathbf{m}}}$ with $t_m = 0.5$. Note that the *pattern error* only increases from 0 to 1 pixel due to the above step. We also mention that the inclusion of the regularization penalty that promotes binary results did not introduce any new error when applied to the pattern in Fig. 6 and, therefore, due to space constraints, we do not show the results for that case.

B. Complexity Penalty Term

A second penalty term we will incorporate will help ensure that the resulting OPC mask is less complex and, therefore, cheap and easy to fabricate and inspect. Isolated perturbations, protrusions, etc., are not preferred because they increase the data handling and manufacturing cost. Hence, we seek a penalty term which suppresses these effects. To achieve this, let us first define an auxiliary variable called *the activation pattern* $\underline{\mathbf{f}}$ where

$$f_j = \left|m_j - z_j^*\right| \quad \text{for} \quad j = 1, \ldots, N^2.$$

The *on* pixels in $\underline{\mathbf{f}}$ indicate the positions where prewarping occurred; so the prewarped pattern can be obtained by simply flipping the corresponding pixels in $\underline{\mathbf{z}}^*$ from 1 to 0 or 0 to 1.

There are a variety of penalty terms that one can employ depending upon how one defines *mask complexity*. Akin to the idea of total variation (TV) [37] penalty, we choose to penalize the mask complexity using the local variation of the activation pattern as follows [38]:

$$\|\nabla\underline{\mathbf{f}}\|_1 = \|\mathbf{Q_x}\underline{\mathbf{f}}\|_1 + \|\mathbf{Q_y}\underline{\mathbf{f}}\|_1 \tag{19}$$

where $\mathbf{Q}_{\rm x}, \mathbf{Q}_{\rm y} \in \Re^{N^2}$ represent the first (directional) derivatives and are defined as $\mathbf{Q_x} = \mathbf{I} - \mathbf{S_x}$ and $\mathbf{Q_y} = \mathbf{I} - \mathbf{S_y}$ where $\mathbf{S_x}$ and $\mathbf{S_y}$ shift the 2-D mask represented by $\underline{\mathbf{m}}$ along (right) horizontal and (up) vertical direction by one pixel, respectively.

This approach, while relatively simple, enables us to decouple the features of the underlying prescribed pattern $\underline{\mathbf{z}}^*$ from $\underline{\mathbf{m}}$ thereby capturing only the *changes* occurring due to prewarping. Isolated holes, protrusions, and jagged edges have higher associated penalty. The regularization term in (19) suppresses these effects and forces the *changes* to be spatially smoother and less abrupt. This leads to simple, and easy to manufacture OPC masks.

The gradient $\nabla R_{\rm TV}(\underline{\mathbf{m}}) \in \Re^{N^2 \times 1}$ of the TV penalty term is given as

$$\nabla R_{\rm TV}(\underline{\mathbf{m}}) = \left[\mathbf{Q_x}^T \text{sign}(\mathbf{Q_x}\underline{\mathbf{f}}) + \mathbf{Q_y}^T \text{sign}(\mathbf{Q_y}\underline{\mathbf{f}})\right] \odot \text{sign}(\underline{\mathbf{m}} - \underline{\mathbf{z}}^*) \tag{20}$$

which can be used in conjunction with (13) and (14) while carrying out the steepest-descent iterations as before.

Fig. 13 illustrates the results with the same prescribed pattern, PSF, and sigmoidal parameters used in Fig. 6 (Section III-B).

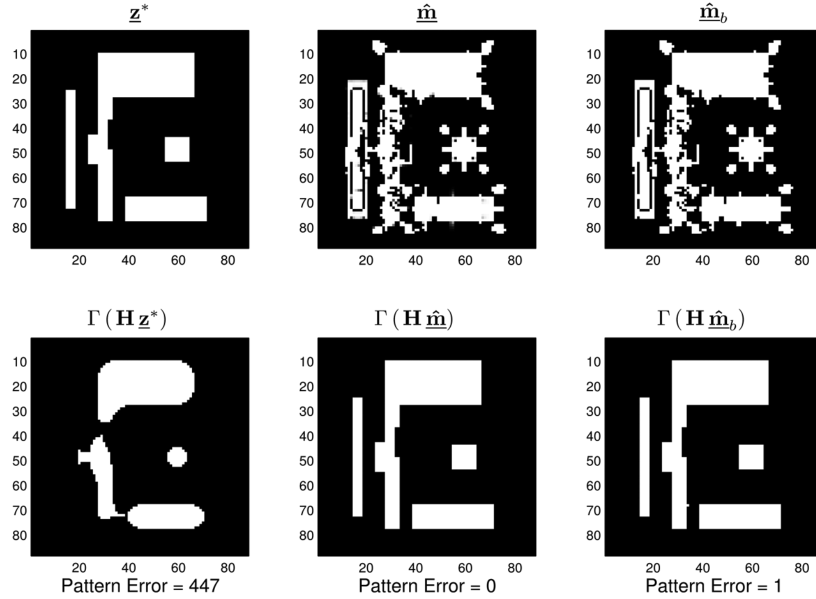

Fig. 12. Left to right: Original pattern, optimized gray pattern, and binary pattern obtained using global thresholding with $t_m = 0.5$. The cost function minimized was $F(\underline{\mathbf{m}}) + \gamma_{\text{bin}} R_{\text{bin}}(\underline{\mathbf{m}})$ and the parameters are $a = 80$, $t_r = 0.5$, 11×11 filter with $\sigma = 14$ pixels, $s = 1.5$, and number of iterations $= 200$, and $\gamma_{\text{bin}} = 0.015$.

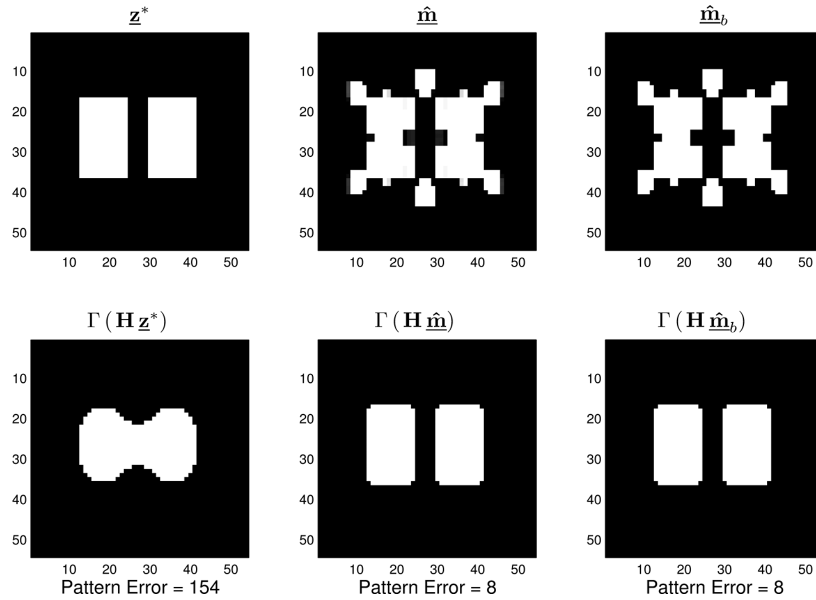

Fig. 13. Left to right: Original pattern, optimized gray pattern, and binary pattern obtained using global thresholding with $t_m = 0.5$. The cost function minimized was $F(\underline{\mathbf{m}}) + \gamma_{\text{bin}} R_{\text{bin}}(\underline{\mathbf{m}}) + \gamma_{\text{TV}} R_{\text{TV}}(\underline{\mathbf{m}})$ and the parameters are $a = 90$, $t_r = 0.5$, 15×15 filter with $\sigma = 5$ pixels, $s = 1$, and number of iterations $= 200$, $\gamma_{\text{bin}} = 0.025$, and $\gamma_{\text{TV}} = 0.045$.

However, here, we employ both the quadratic and TV regularization terms and minimize the overall cost function

$$J(\underline{\mathbf{m}}) = F(\underline{\mathbf{m}}) + \gamma_{\text{bin}} R_{\text{bin}}(\underline{\mathbf{m}}) + \gamma_{\text{TV}} R_{\text{TV}}(\underline{\mathbf{m}})$$

where $F(\underline{\mathbf{m}})$ is defined in (8), $R_{\text{bin}}(\underline{\mathbf{m}})$ is defined in (17), and $R_{\text{TV}}(\underline{\mathbf{m}})$ is defined in (19) with $\gamma_{\text{bin}} = 0.025$ and $\gamma_{\text{TV}} = 0.045$. The masks were obtained using 200 iterations with $s = 1.0$. We observe that the synthesized OPC pattern is much simpler than the one in Fig. 6, yet the *pattern error* is still quite small.

Similarly, Fig. 14 illustrates the results obtained using both regularization terms with the same prescribed pattern, PSF, and sigmoidal parameters used in Fig. 12 (Section IV-A) with $\gamma_{\text{bin}} = 0.01$ and $\gamma_{\text{TV}} = 0.02$. The above result was obtained in 400 iterations with $s = 1$. In both of the above cases, we observe that the isolated holes, protrusions, and jagged edges are suppressed, and, hence, the resulting mask is simpler, and, therefore, easier, and cheaper to fabricate and inspect.

C. More Results

In this section, we present two interesting results using square and circular lens apertures.

1) Comparison With Past Work: Here we compare the results obtained using our continuous function optimization method

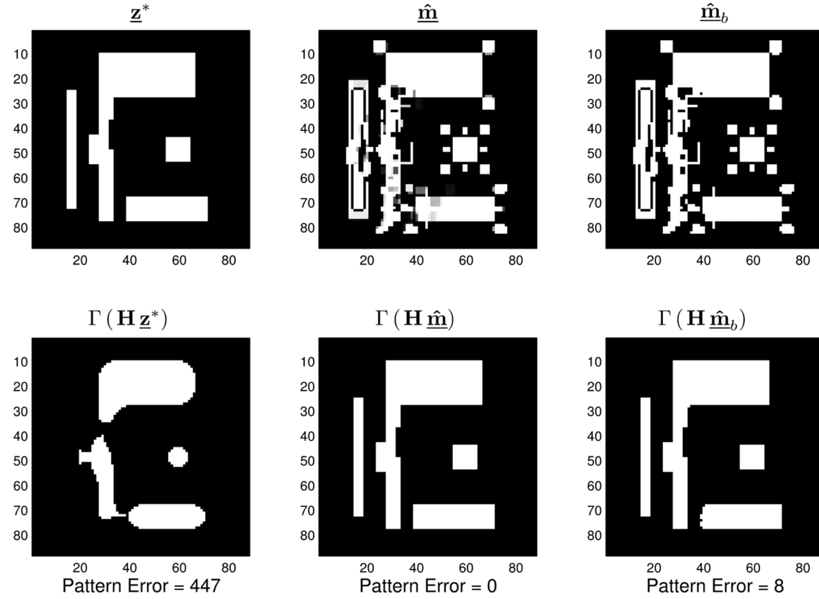

Fig. 14. Left to right: Original pattern, optimized gray pattern, and binary pattern obtained using global thresholding with $t_m = 0.5$. The cost function minimized was $F(\underline{\mathbf{m}}) + \gamma_{\rm bin} R_{\rm bin}(\underline{\mathbf{m}}) + \gamma_{\rm TV} R_{\rm TV}(\underline{\mathbf{m}})$ and the parameters are $a = 80$, $t_r = 0.5$, 11×11 filter with $\sigma = 14$ pixels, $s = 1$, number of iterations $= 400$, $\gamma_{\rm bin} = 0.01$, and $\gamma_{\rm TV} = 0.02$.

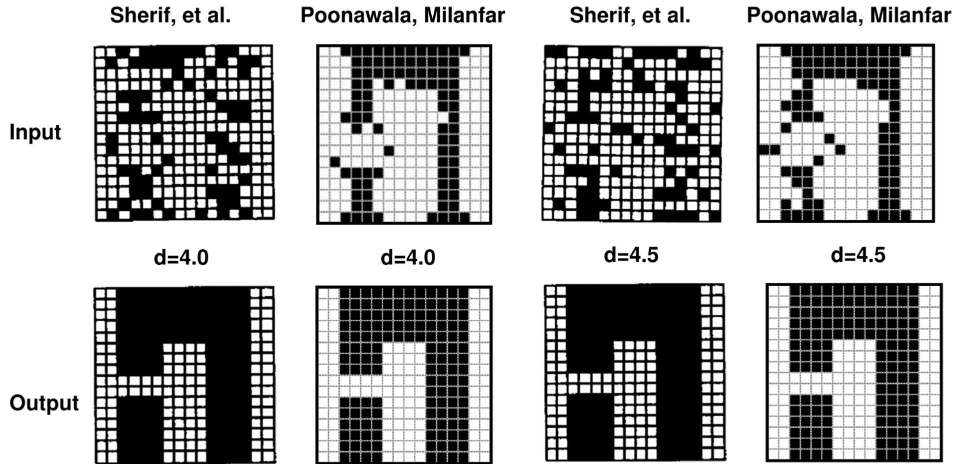

Fig. 15. Top row indicates the synthesized masks obtained using Sherif *et al.* (from [24]) and our algorithm for a square pupil with $d = 4.0$ and $d = 4.5$ pixels. The bottom row indicates the corresponding output wafer patterns assuming a hard-limiter with $t_r = 0.2$.

against those obtained by integer optimization. Sherif *et al.* [24] employed mixed-linear integer programming to estimate the resist threshold and the OPC mask for an idealized lithography system with a square aperture. Their cost function is very similar to the *pattern error* defined in (15) (note that we have used the latter as a metric to evaluate the quality of our results). The authors in [24] reduce the nonlinear cost function to a linear objective function with unconditional constraints by tripling the number of estimated variables. The problem can then be solved using branch and bound method.

In order to make a fair comparison, we choose the same imaging system and target pattern ($\underline{\mathbf{z}}^*$) as employed by the authors in [24]. The imaging systems considered in [24] is an incoherent imaging system with a square aperture. In this case, the PSF is defined as

$$h_{p,q} = \frac{1}{d^2}\text{sinc}^2\left(\frac{p}{d}\right)\text{sinc}^2\left(\frac{q}{d}\right)$$

where d is the width of the impulse response function in units of pixels. The results for the cases of $d = 4.0$ and $d = 4.5$ are illustrated in Fig. 15. The experimental parameters for our algorithm are as follows: $a = 60$, $t_r = 0.2$, and $\gamma_{\rm bin} = 0.03$. We observe that the synthesized masks obtained using our algorithm are different from those obtained by solving the combinatorial optimization problem. However, the output wafer pattern is reproduced with 100% accuracy proving that our results are correct. It is also worth noting that our masks were synthesized in 5 s using a 1.4-GHz Pentium-M machine compared to 15–25 min reported in [24] (using the computational resources available in 1995).

2) Circular Lens Aperture: Fig. 16 illustrates the results for an incoherent imaging system having circular lens aperture with $NA = 0.95$ and $\lambda = 193$ nm. The PSF for the above system is the square of a jinc function with cut-off frequency NA/λ [16], [39]. The desired pattern is sampled at 5 nm and consists

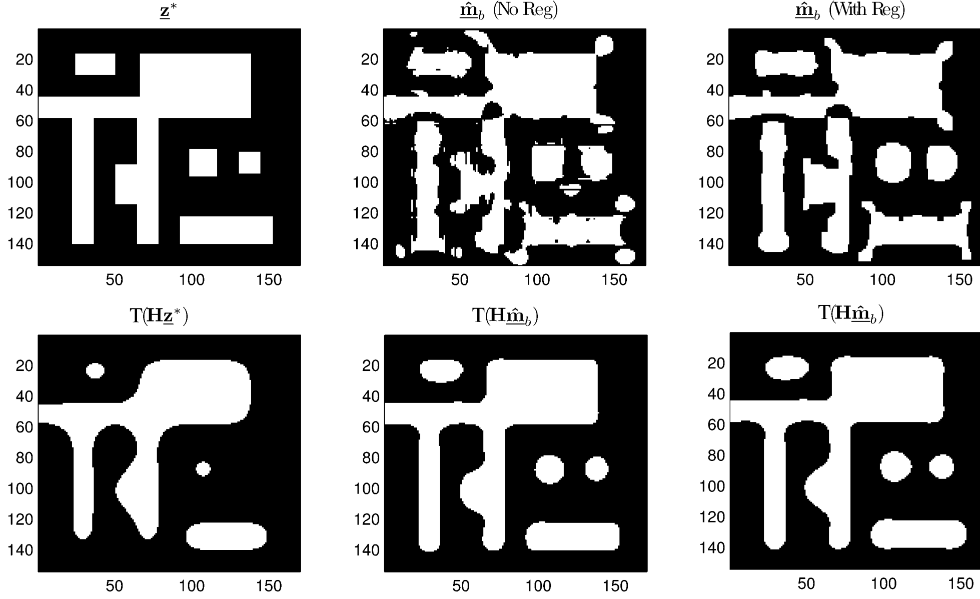

Fig. 16. Left to right: Desired pattern, the synthesized binary mask using no regularization ($a = 40, t_r = 0.5$), and the synthesized binary mask with $\gamma_{\text{bin}} = 0.01$ and $\gamma_{\text{TV}} = 0.01$ for a circular pupil with $k_1 = 0.34$.

of 70-nm features giving $k_1 = 70 * 0.95/193 = 0.34$. The top row in Fig. 16 indicates the input masks and the bottom row indicates the corresponding wafer patterns. We observe that the synthesized mask significantly reduces the corner rounding and brings the contours on target. It is also interesting to note that the algorithm automatically decided to *break* some features into two disjoint parts. For example, the two vertical bars are disconnected from the main feature in the synthesized mask, but are accurately reproduced at the output. Such counter-intuitive results are hard to obtain using edge-based parametrization and would require extensive and tedious segmentation scripts. Interestingly, the above feature-breaking behavior was also observed very recently by Liu *et al.* (see [34, Fig. 4]).

V. Weight Mask and Selected Parameter Optimization

In this section, we discuss two additional useful features which can be incorporated in the above framework for solving the OPC mask design problem.

A. Weight Mask

In many practical scenarios, it is of interest to reproduce certain regions or areas of the pattern more faithfully as compared to others. Consider two parallel wires; the key requirement is that they never fuse on the reproduced pattern, whereas we are much less concerned about their actual reproduced shape. Similarly the mask designer can mark other critical regions depending on the shape, size and vicinity of the neighboring features. The above information can be incorporated into a *weight pattern* which can then be used for selective region correction. The goal is that only the critically marked (high priority) regions on the desired pattern should be faithfully replicated using suitable prewarping. The unimportant (low weight) regions of the underlying pattern may be reproduced with relatively lower fidelity. The weight pattern is assigned before running the OPC routine and can be generated heuristically or by the mask designer.

Let $\underline{\mathbf{w}} \in \Re^{N^2 \times 1}$ represent the weight mask where each element $w_i \in [0, 1]$ signifies the importance (or weight) associated with reproducing that particular pixel at the output. We reformulate the optimization problem in (9) using the weighted cost function as follows:

$$\hat{\underline{\mathbf{m}}} = \arg\min_{\underline{\mathbf{m}}} \sum_{i=1}^{N^2} w_i \left(z_i^* - z_i\right)^2. \tag{21}$$

The gradient calculations and steepest-descent iterations can be easily extended using (13) and (14) as follows:

$$\underline{\boldsymbol{\theta}}^{k+1} = \underline{\boldsymbol{\theta}}^k - s\left\{a\mathbf{H}^{\mathbf{T}}\left[\underline{\mathbf{w}}\odot(\underline{\mathbf{z}}^* - \underline{\mathbf{z}})\odot\underline{\mathbf{z}}\odot(1-\underline{\mathbf{z}})\right]\odot\sin(\underline{\boldsymbol{\theta}}^k)\right\}. \tag{22}$$

Fig. 17 illustrates the weight mask (left), optimized OPC pattern (center), and the final output pattern (right) using the same parameters as in Fig. 16 with $\gamma_{\text{bin}} = 0.02$ and $\gamma_{\text{TV}} = 0.04$. In this case, we are not concerned about reproducing the corners accurately at the output. However, we want the other features of the pattern (length, width and separation of the bars) to be accurately reproduced. The above objective can be achieved by incorporating the information from the weight mask in Fig. 17. Note that a 5×5 region around every corner has been assigned zero weight, whereas the remaining regions have been assigned a higher weight equal to one. The resulting prewarping is different from that occurring in Fig. 6 and all the regions, barring corners, are perfectly reproduced.

B. Selected Parameter Optimization

All the algorithms discussed up to this point treat the entire synthesized pattern to be completely unknown. The user (mask designer) has no control over which regions of the underlying pattern are allowed to be predistorted. The entire layout space

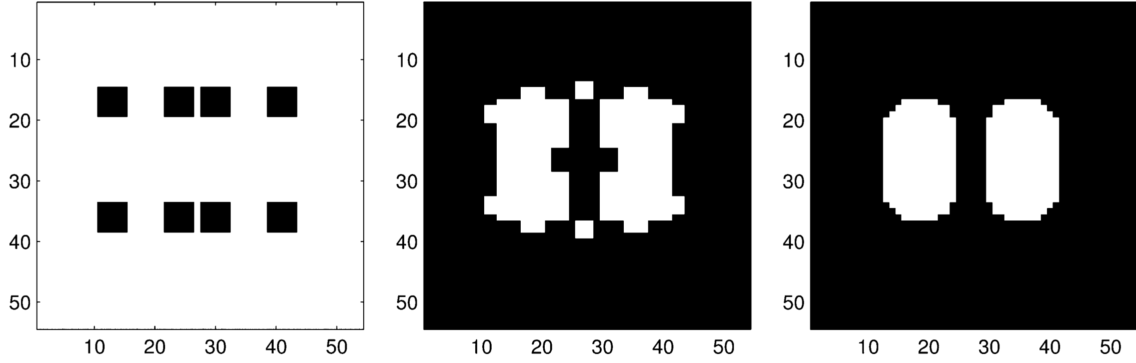

Fig. 17. (Left) Employed weight mask, (center) synthesized mask pattern, and (right) the final output pattern. Note that 5 × 5 region around the corners have zero weight whereas the weight is one everywhere else.

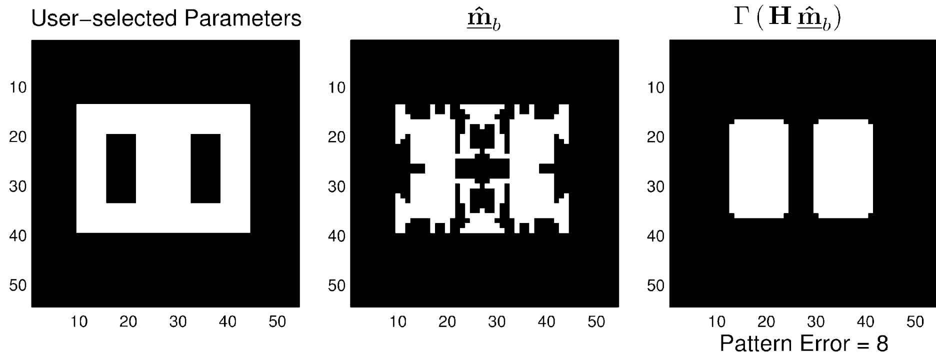

Fig. 18. Left to right: User-selected parameters (marked in white), the corresponding optimized mask pattern, and the final output pattern. The parameters are $\mathrm{a} = 90, t_r = 0.5$, 15 × 15 filter with $\sigma = 5, \lambda_{bin} = 0.01$, and $\mathrm{s} = 1$.

is not always at our disposal and some design rules may require us to force our variations to fall only within a specified area or region of the synthesized mask. In this section, we outline an approach which provides the above flexibility. This may also be beneficial in saving the computational time required to reconfigure OPC solutions in the event of design re-spins, errors, etc. Note that the focus here is on controlling the areas of the input that undergo changes, whereas in Section V-A, the focus was on controlling the areas of the output to be accurately reproduced.

The procedure starts with the mask designer (or the OPC engineer) marking (or selecting) regions (or pixels) in the original pattern where predistortion is permissible. Alternatively, the above procedure can also be automated based on feature type, density, vicinity, etc., and incorporated into the CAD software. The above pixels are stacked together to form the unknown parameter $\underline{\mathbf{m}}_{\mathbf{1}} \in \Re^{P\times 1}$. The remaining pixels will stay unchanged in the synthesized mask and are stacked together to obtain the known vector $\underline{\mathbf{m}}_{\mathbf{2}} \in \Re^{(N^2-P)\times 1}$.

Note that an element h_{ij} in matrix $\mathbf{H}$ denotes the contribution of the jth pixel towards the formation of the output z_i. The above matrix can be rewritten as $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_{N^2}}]$ where $\mathbf{h_k} \in \Re^{N^2\times 1}$ represents the contribution of the kth pixel towards the formation of the output vector $\underline{\mathbf{z}}$. The $\mathbf{h_k}$s corresponding to the pixels in $\underline{\mathbf{m}}_{\mathbf{1}}$ are stacked together to form $\mathbf{H_1} \in \Re^{N^2\times P}$ and the remaining $\mathbf{h_k}$s are stacked to form $\mathbf{H_2} \in \Re^{N^2\times (N^2-P)}$. Thus, $\mathbf{H}$ is split into two components where $\mathbf{H_1}$ determines the contribution from pixels which are unknown (allowed to be modified) and $\mathbf{H_2}$ determines the contribution from pixels which are fixed. The forward model in (6) can now be rewritten as

$$\underline{\mathbf{z}} = \mathrm{sig}(\mathbf{H_1}\underline{\mathbf{m}}_{\mathbf{1}} + \mathbf{H_2}\underline{\mathbf{m}}_{\mathbf{2}}). \tag{23}$$

Note that the second summation term inside the sigmoid is a constant because $\underline{\mathbf{m}}_{\mathbf{2}}$ remains unchanged. Finally, the steepest descent iterations can be directly used to update the modifiable pixels in $\underline{\mathbf{m}}_{\mathbf{1}}$ as follows:

$$\underline{\boldsymbol{\theta}}_1^{k+1} = \underline{\boldsymbol{\theta}}_1^{k} - s\Big\{a\left(\mathbf{H_1^T}\left[(\underline{\mathbf{z}}^{*} - \underline{\mathbf{z}}) \odot \underline{\mathbf{z}} \odot (\mathbf{1} - \underline{\mathbf{z}})\right]\right) \odot \sin\left(\underline{\boldsymbol{\theta}}_1^{k}\right)\Big\}. \tag{24}$$

Fig. 18 illustrates a user-defined band (three pixel thick on both side of the edges) indicating the region allowed to undergo predistortion. The user-selected region is marked in white, and the remaining region where predistortion is nonpermissible is marked in black. The chosen parameters were used to solve for the same experiment and pattern as in Fig. 6 albeit with $\gamma_{\mathrm{bin}} = 0.01$ and $s = 1$. Fig. 18 also illustrates the optimized binary OPC pattern and the final binary output pattern using the above method. As expected, the changes are confined to lie within the band and are very different (far less scattered) compared to the synthesized OPC mask in Fig. 6.

VI. Extensions, Other Applications, and Conclusion

In this section, we discuss some extensions and future direction which will make our work more beneficial to the lithography community. We also provide some concluding remarks.

A. Coherent and Partially Coherent Imaging Systems

Practical real-world lithography systems are partially coherent imaging systems. Therefore, the inverse imaging problem should be solved for the above case to have practical value. In this section, we discuss how our proposed framework could be possibly extended to synthesize masks for fully coherent and partially coherent imaging systems.

1) Coherent Imaging System: In the case of coherent imaging system, the amplitude of the output electric field is linearly related to the input electric field amplitude generated by the mask $m(x, y)$. The photo-resist responds to the intensity of the electric field, where intensity is defined as the square of the amplitude. Therefore, the forward model is defined as

$$\underline{\mathbf{z}}^{\mathbf{c}} = \mathrm{sig}\left(|\mathbf{H}\underline{\mathbf{m}}|^2\right). \quad (25)$$

The sigmoid function simulates the resist behavior and acts on the aerial image $|\mathbf{H}\underline{\mathbf{m}}|^2$ (square of the amplitude), giving the output photo-resist pattern $\underline{\mathbf{z}}^{\mathbf{c}}$. Note that $|\cdot|^2$ operator here implies element-by-element absolute square of the individual vector entries. Typical lithography systems employ a circular lens aperture, where the coherent imaging system now acts as an ideal low pass filter. Therefore, the convolution kernel $\mathbf{H}$ is a jinc function with cut-off frequency NA/λ. For partially coherent imaging systems, the optical kernel $h(x, y)$ can be instead substituted by the optimal coherent approximation proposed by Pati and Kailath [26].

2) Partially Coherent Imaging System: Partially coherent imaging systems are modelled using the Hopkins diffraction model [16]. Pati and Kailath [26] proposed an approximation to the above model called the sum-of-coherent-system (SOCS) by using the singular value decomposition of the transmission cross-coefficient matrix. In their approach, the Pth order approximation to the aerial image formulation can be calculated using the weighted sum of P coherent systems. The forward model now becomes

$$\underline{\mathbf{z}}^{\mathbf{P}} = \mathrm{sig}\left(\sum_{j=1}^{P} \sigma_j |\mathbf{H}_j \underline{\mathbf{m}}|^2\right) \quad (26)$$

where $\mathbf{H}_j$ for $j = 1, \ldots, P$ are the amplitude spread functions (also referred to as optical system kernels) of the coherent systems, and $\sigma_1, \ldots, \sigma_P$ are the corresponding singular values. The singular values quickly decay to zero, thereby facilitating an accurate reduced order approximation.

The optimization problem can be formulated as the L_2 norm of the difference between $\underline{\mathbf{z}}^*$ and $\underline{\mathbf{z}}^{\mathbf{c}}$ (or $\underline{\mathbf{z}}^{\mathbf{P}}$) similar to (8) which can be solved using our algorithm. All the elements of our proposed framework such as the formulation of a continuous function optimization, analytic gradient calculation, parametric transformation, and regularization can also be employed to solve for the above imaging systems. The analytic gradient calculation would require $O(N^2 \log(N))$ and $O(PN^2 \log(N))$ operations for the coherent and partially coherent imaging systems respectively. Preliminary results using coherent imaging system to synthesize binary and phase shift masks were recently reported in [40]. Thus, we believe that our framework is generic enough to account for the above cases.

B. Extensions in Microlithography

In this paper, we discussed the OPC mask synthesis problem for an idealized microlithography system. An important extension (already discussed above) is to account for partially coherent imaging systems. With reducing values of k_1, scattered light also referred to as *flare* is becoming an important contributor to undesired CD variations [41]. Flare is lithography tool specific. Therefore, the knowledge of the tool used for mask exposure is also required. The forward model can be augmented to incorporate the flare effects and we can then synthesize the OPC mask to solve for both proximity and flare-induced errors for the specified tool. Practical lithography systems are also prone to random uncontrollable process errors causing focus and exposure variations. These lead to loss in pattern fidelity and reduction in yield. An important future direction would be to synthesize masks which improve the robustness of the lithography process. Another very promising extension is to synthesize phase shift masks where the estimated pixels now also have a phase information associated with them. The proposed framework will enable us to combine automated phase shift and OPC mask design, a powerful combination of RETs [4].

Other attractive possibilities include maskless lithography optimization [42] and diffractive optical element (DOE) design for customized illumination [43]. Scaling our algorithm to full-chip level forms another important direction. The computational complexity for estimating N^2 pixels is $O(N^2 \log(N))$. Therefore, as the size of the pattern increases, the run-time of our algorithm scales in accordance to the above formula. For full-chip level, there is a possibility of splitting the entire die into smaller patterns, solving the individual subproblems in parallel, and finally them stitching them together as observed in [44]. Furthermore, explicit mask-manufacturing constraints like fracture dimensions and minimum feature size/spacing are not considered in our problem formulation; hence, the resulting masks may not always be manufacturable. However, the regularization framework can be extended to incorporate the above as observed by Pang *et al.* in [45].

The proposed approach may also be beneficial to electron beam lithography. Here, the desired pattern is directly written by firing a focused beam of electrons onto a photoresist-coated substrate. This process suffers distortions known as proximity effects arising from forward and back-scattering of electrons, beam density blurring, etc. The above effect is counteracted using electron-beam proximity correction, a predistortion technique commonly adopted by the nanolithography community [46]. Our proposed framework can be adopted to modify the dose and/or shape of the input patterns in order to obtain a higher pattern fidelity. However, the success of our method depends on the availability of a faithful forward model which accurately describes the e-beam lithography process.

C. Conclusion

We demonstrated the application of well-studied tools for image restoration like regularization and nonlinear optimization to solve the inverse lithography problem for an incoherent imaging system. The novelty of our method lies in the fact that we used the continuous function formulation and use the gradient information to systematically explore the solution

space. The latter is calculated analytically having complexity $O(N^2 \log(N))$ where N^2 is the total number of pixels. A regularization framework was successfully used to control the tone and complexity of the synthesized OPC masks. The results demonstrated interesting feature breaking for low k_1 values. The results were also compared with those obtained using integer programming and were found to be equally good. Our framework needs to be extended to partially coherent imaging systems to solve the real-world microlithography problem.

With the critical dimensions hitting 45 nm, the semiconductor industry is striving to find aggressive RET solutions. This has led to a recent revival of interest in pixel-based RET [27], [29], [34], [44]. The algorithms and framework presented in this paper are promising and will contribute towards that direction.

APPENDIX
GRADIENT DERIVATION

Here, we present the derivation for the gradient of the cost function in (12). First, we rewrite (12) as

$$F_1(\underline{\boldsymbol{\theta}}) = \sum_{i=1}^{N^2} (z_i^* - z_i(\underline{\boldsymbol{\theta}}))^2 \tag{27}$$

where

$$z_i(\underline{\boldsymbol{\theta}}) = \frac{1}{1 + \exp\left[-a\left(\sum_{j=1}^{N^2} h_{ij} \frac{1+\cos(\theta_j)}{2}\right) + at_r\right]}.$$

The partial derivative of (27) with respect to θ_k is given as

$$\begin{aligned}
\frac{\partial F_1(\underline{\boldsymbol{\theta}})}{\partial \theta_k}
&= 2\sum_{i=1}^{N^2} (z_i^* - z_i(\underline{\boldsymbol{\theta}}))\left(-\frac{\partial z_i(\underline{\boldsymbol{\theta}})}{\partial \theta_k}\right) \\
&= 2\sum_{i=1}^{N^2} (z_i^* - z_i(\underline{\boldsymbol{\theta}}))\, z_i^2(\underline{\boldsymbol{\theta}}) \\
&\quad \frac{\partial}{\partial \theta_k} \exp\left(-a\sum_{j=1}^{N^2} h_{ij}\frac{1+\cos(\theta_j)}{2} + at_r\right) \\
&= 2\sum_{i=1}^{N^2} (z_i^* - z_i(\underline{\boldsymbol{\theta}}))\, z_i(\underline{\boldsymbol{\theta}})\,(1 - z_i(\underline{\boldsymbol{\theta}})) \\
&\quad \frac{\partial}{\partial \theta_k}\left(-a\sum_{j=1}^{N^2} h_{ij}\frac{1+\cos(\theta_j)}{2} + at_r\right) \\
&= a\sum_{i=1}^{N^2} (z_i^* - z_i(\underline{\boldsymbol{\theta}}))\, z_i(\underline{\boldsymbol{\theta}})\,(1 - z_i(\underline{\boldsymbol{\theta}}))\sin(\theta_k) h_{ik} \\
&= a\sin(\theta_k)\sum_{i=1}^{N^2} h_{ik}\,(z_i^* - z_i(\underline{\boldsymbol{\theta}}))\, z_i(\underline{\boldsymbol{\theta}})\,(1 - z_i(\underline{\boldsymbol{\theta}})).
\end{aligned}$$

Therefore, the gradient vector can be calculated as

$$\begin{aligned}
\nabla F_1(\underline{\boldsymbol{\theta}}) &= \left[\frac{\partial F_1(\underline{\boldsymbol{\theta}})}{\partial \theta_1}, \ldots, \frac{\partial F_1(\underline{\boldsymbol{\theta}})}{\partial \theta_{N^2}}\right]^T \\
&= a\left(\mathbf{H^T}\left[(\underline{\mathbf{z}}^* - \underline{\mathbf{z}}) \odot \underline{\mathbf{z}} \odot (1 - \underline{\mathbf{z}})\right]\right) \odot \sin(\underline{\boldsymbol{\theta}}).
\end{aligned}$$

The above can be evaluated directly using 2-D matrices as demonstrated in Fig. 5.

REFERENCES

[1] S. Sayegh and B. Saleh, "Image design: Generation of a prescribed image at the output of a bandlimited system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 4, pp. 441–445, Jul. 1983.

[2] S. Sayegh, B. Saleh, and K. Nashold, "Image design: Generation of a prescribed image through a diffraction limited system with highcontrast recording," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 460–465, Apr. 1985.

[3] R. Gonzalez and R. Woods, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[4] F. Schellenberg, "A little light magic," *IEEE Spectrum*, vol. 40, no. 9, pp. 34–39, Sep. 2003.

[5] P. Choudhury, *Handbook of Microlithography, Micromachining and Microfabrication*. Bellingham, WA: SPIE, 1997.

[6] ITRS, 2006, International Technology Roadmap for Semiconductors [Online]. Available: http://public.itrs.net/

[7] S. Owa and H. Nagasaka, "Immersion lithography; its potential performance and issues," in *Proc. SPIE Optical Microlithography*, 2003, vol. 5040, pp. 724–733.

[8] F. Schellenberg, *Resolution Enhancement Techniques in Optical Lithography*. Bellingham, WA: SPIE, 2004.

[9] L. Liebmann, S. Mansfield, A. Wong, M. Lavin, W. Leipold, and T. Dunham, "TCAD development for lithography resolution enhancement," *IBM J. Res. Develop.*, vol. 45, pp. 651–665, 2001.

[10] A. K.-K. Wong, *Resolution Enhancement Techniques in Optical Lithography*. Bellingham, WA: SPIE, 2001.

[11] S. Hsu, J. Park, D. Broeke, and J. Chen, "Double exposure technique for 45 nm and beyond," presented at the SPIE BACUS Symp. Photomask Technology, 2005.

[12] F. Schellenberg, "Resolution enhancement with OPC/PSM," *Future Fab. Int.* vol. 9, 2000.

[13] O. Otto, J. Garofalo, K. Low, C. Yuan, R. Henderson, C. Pierrat, R. Kostelak, S. Vaidya, and P. Vasudev, "Automated optical proximity correction—A rules-based approach," in *Proc. SPIE Optical Microlithography*, 1994, vol. 2197, pp. 278–293.

[14] N. Cobb and Y. Granik, "Dense OPC for 65 nm and below," presented at the SPIE BACUS Symp. Photomask Technology, 2005.

[15] A. Erdmann and W. Henke, "Simulation of optical lithography," in *Proc. SPIE Optical Microlithography*, 1999, vol. 3729, pp. 480–494.

[16] M. Born and E. Wolfe, *Principles Opt.*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[17] N. Cobb and A. Zakhor, "Fast sparse aerial image calculation for OPC," in *Proc. SPIE BACUS Symp. Photomask Technology*, 1995, vol. 2621, pp. 534–545.

[18] C. Mack, "Development of positive photoresist," *J. Electrochem. Soc.*, vol. 134, pp. 148–152, 1987.

[19] S. Robertson, C. Mack, and M. Maslow, "Toward a universal resist dissolution model for lithography simulation," in *Proc. SPIE Lithography for Semiconductor Manufacturing II*, 2001, vol. 4404, pp. 111–122.

[20] W. Huang, C. Lin, C. Kuo, C. Huang, J. Lin, J. Chen, R. Liu, Y. Ku, and B. Lin, "Two threshold resist models for optical proximity correction," in *Proc. SPIE Optical Microlithography*, 2001, vol. 5377, pp. 1536–1543.

[21] J. Randall, K. Ronse, T. Marschner, M. Goethals, and M. Ercken, "Variable threshold resist models for lithography simulation," in *Proc. SPIE Optical Microlithography*, 1999, vol. 3679, pp. 176–182.

[22] C. Ahn, H. Kim, and K. Baik, "Novel approximate model for resist process," in *Proc. SPIE Optical Microlithography*, 1998, vol. 3334, pp. 752–763.

[23] N. Cobb and A. Zakhor, "Fast, low-complexty mask design," in *Proc. SPIE Optical Microlithography*, 1995, vol. 2440, pp. 313–327.

[24] S. Sherif, B. Saleh, and R. Leone, "Binary image synthesis using mixed linear integer programming," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1252–1257, Sep. 1995.

[25] Y. Liu and A. Zakhor, "Binary and phase shifting mask design for optical lithography," *IEEE Trans. Semiconduct. Manufact.*, vol. 5, no. 2, pp. 138–151, May 1992.

[26] V. Pati and T. Kailath, "Phase-shifting masks for microlithography: Automated design and mask requirements," *J. Opt. Soc. Amer. A*, vol. 9, pp. 2438–2452, 1994.

[27] Y. Oh, J. C. Lee, and S. Lim, "Resolution enhancement through optical proximity correction and stepper parameter optimization for 0.12-μm mask pattern," in *Proc. SPIE Optical Microlithography*, 1999, vol. 3679, pp. 607–613.

[28] A. Erdmann, R. Farkas, T. Fuhner, B. Tollkuhn, and G. Kokai, "Towards automatic mask and source optimization for optical lithography," in *Proc. SPIE Optical Microlithography*, 2004, vol. 5377, pp. 646–657.

[29] Y. Granik, "Solving inverse problems of optical microlithography," in *Proc. SPIE Optical Microlithography*, 2005, vol. 5754, pp. 506–526.

[30] A. Poonawala and P. Milanfar, "Prewarping techniques in imaging: Applications in nanotechnology and biotechnology," in *Proc. SPIE Electronic Imaging*, 2005, vol. 5674, pp. 114–127.

[31] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Comput. Surv.*, vol. 2, pp. 163–213, 1999.

[32] Y. Liu and A. Zakhor, "Optimal binary image design for optical lithography," in *Proc. SPIE Optical Microlithography*, 1990, vol. 1264, pp. 401–412.

[33] K. Nashold and B. Saleh, "Image construction through diffraction-limited high contrast imaging system: An iterative approach," *J. Opt. Soc. Amer. A*, vol. 2, pp. 635–643, 1985.

[34] Y. Liu, D. Abrams, L. Pang, and A. Moore, "Inverse lithography technology principles in practice: Unintuitive patterns," in *Proc. BACUS Symp. Photomask Technology*, 2005, vol. 5992, pp. 231–238.

[35] M. Bazaraa and C. Shetty, *Non-Linear Programming Theory and Algorithms*. New York: Wiley, 1979.

[36] C. Kelly, *Iterative Methods for Optimization*. Philadelphia, PA: SIAM, 1999.

[37] C. Vogel, *Computational Methods for Inverse Problems*. Philadelphia, PA: SIAM, 2002.

[38] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super-resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[39] R. Wilson, *Fourier Series and Optical Transform Techniques in Contemporary Optics*. New York: Wiley, 1995.

[40] A. Poonawala and P. Milanfar, "OPC and PSM design using inverse lithography: A non-linear optimization approach," in *Proc. SPIE Optical Microlithography*, 2006, vol. 6154, pp. 114–127.

[41] J. Word, J. Belledent, Y. Trouiller, Y. Granik, O. Toublan, and W. Maurer, "Full chip model based correction of flare-induced linewidth variation," in *Proc. SPIE Optical Microlithography*, 2005, vol. 5754, pp. 1209–1219.

[42] T. Sandstrom and H. Martinsson, "RET for optical maskless lithography," in *Proc. SPIE Optical Microlithography*, 2004, vol. 5377, pp. 1750–1763.

[43] Y. Granik, "Illuminator optimization methods in microlithography," in *Proc. SPIE Optical Microlithography*, 2004, vol. 5524, pp. 217–229.

[44] B. Lin, M. Shieh, J. Sun, J. Ho, Y. Wang, X. Wu, W. Leitermann, O. Lin, J. Lin, Y. Liu, and L. Pang, "Inverse lithography technology at chip scale," presented at the SPIE Optical Microlithography XIX, 2006.

[45] L. Pang, Y. Liu, and D. Abrams, "Inverse lithography technology (ILT), what is the impact on the photomask industry?," presented at the Symp. Photomask and Next Generation Mask Technology VII, 2006.

[46] B. Cook, "PYRAMID—A hierarchical rule-based proximity effect correction scheme for electron-beam lithography: Generalization and optimization for homogeneouse substrates," Ph.D. dissertation, Cornell Univ., New York, 1996.

Amyn Poonawala received the B.E. degree from the University of Mumbai, India, in 2001, and the M.S. degree from the University of California, Santa Cruz (UCSC), in 2004, both in computer engineering. He is currently pursuing the Ph.D. degree in the Computer Engineering Department, UCSC.

His technical interests include inverse problems in imaging, particularly their applications for single- and double-exposure mask design in optical microlithography.

Peyman Milanfar (SM'98) received the B.S. degree in electrical engineering and mathematics from the University of California, Berkeley, in 1988, and the S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1993, respectively.

Until 1999, he was a Senior Research Engineer at SRI International, Menlo Park, CA. He is currently an Associate Professor of electrical engineering, University of California, Santa Cruz. He was a Consulting Assistant Professor of computer science at Stanford University, Stanford, CA, from 1998 to 2000, where he was also a Visiting Associate Professor in 2002. His technical interests are in statistical signal and image processing and inverse problems.

Dr. Milanfar won a National Science Foundation CAREER award in 2000, and he was an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS from 1998 to 2001.