

Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis

Ningning Jia and Edmund Y Lam

Imaging Systems Laboratory, Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong

E-mail: elam@eee.hku.hk

Received 22 January 2010, accepted for publication 4 March 2010

Published 1 April 2010

Online at stacks.iop.org/JOpt/12/045601

Abstract

Inverse lithography technology (ILT) synthesizes photomasks by solving an inverse imaging problem through optimization of an appropriate functional. Much effort on ILT is dedicated to deriving superior masks at a nominal process condition. However, the lower k_1 factor causes the mask to be more sensitive to process variations. Robustness to major process variations, such as focus and dose variations, is desired. In this paper, we consider the focus variation as a stochastic variable, and treat the mask design as a machine learning problem. The stochastic gradient descent approach, which is a useful tool in machine learning, is adopted to train the mask design. Compared with previous work, simulation shows that the proposed algorithm is effective in producing robust masks.

Keywords: inverse imaging, lithography, optical, proximity correction, robustness, stochastic gradient descent, machine learning

(Some figures in this article are in colour only in the electronic version)

1. Introduction

1.1. Inverse lithography

With the continuous shrinkage of the circuit minimum feature size, optical proximity correction (OPC) has been developed as a widely used resolution enhancement technique (RET) [1]. Instead of the edge-moving correction scheme adopted by the conventional OPC, the inverse image synthesis method has been used to calculate optimal masks in certain process conditions, a method termed inverse lithography technology (ILT), since the 1990s [2–5]. ILT optimizes masks unconstrained by the topology of the original design [6], and has shown great promise in meeting the challenges in future technology nodes. Many algorithms on ILT have been proposed in recent years. Granik formulated inverse mask synthesis variously as linear, quadratic and nonlinear problems, and classified methods solving these three problems [7]. Erdmann used a genetic algorithm (GA) to solve the inverse mask and source optimization problem [8]. The level-set method has also been explored as an effective approach [9],

and an analysis on its numerical formulation and details of its implementation have been published recently [10]. Poonawala and Milanfar formulated the mask synthesis problem as a steepest descent optimization [11], based on which additional work has been done to refine its implementation [12–16]. Zhang *et al* proposed an innovative iterative method recently, flipping pixels guided by the first- and second-order derivatives of the objective function [17]. These algorithms enrich the tools available for solving ILT problems.

With a lower k_1 factor, process variations, such as focus and dose variations, have caused more CD variations and hotspots. However, most algorithms mentioned above are designed for a nominal process condition only. Regularization on process robustness and the mask manufacturability [18] are still the main concerns before we can push inverse lithography from ‘virtual virtuality’ to real-world manufacturing [19].

Nevertheless, some efforts have been made to enhance the robustness to process variations. Multiple process conditions are incorporated into current OPC recipes [20–23]. A defocus aerial image model is proposed, but the analytical formulation is only applicable on conventional edge-based OPC [24]. An

optimization framework for robust mask design with defocus has been proposed [25] which optimizes the expectation of cost functions under different defocus conditions by taking several defocus samples. This numerical approach can generate masks with superior performance in a range of defocus conditions over those optimized at best focus. However, it requires more computation, and is proportional to the sample size. In this paper, we treat the mask optimization as a training process, and adopt the stochastic gradient descent approach to optimize more efficiently the mask with process robustness.

1.2. Stochastic gradient descent in machine learning

Machine learning is concerned with constructing programs that automatically improve their behavior with experience [26]. In this work, we model the inverse mask synthesis with process variations as a general machine learning problem. This approach has been used in optical lithography, for example, for hotspot detection [27] and variability prediction [28]. Neural networks have also been applied to electron-scattering proximity correction [29] and one-step rule-based OPC [30]. Recently, linear regression was used for a computationally efficient OPC in the conventional edge-moving recipe [31]. However, in this paper for the *first* time, as far as we know, inverse lithography is formulated into a machine learning model for the purpose of solving the robust mask design problem.

In machine learning, the objectives to be learned can be functions, logic programs and rule sets, grammars, or problem solving systems [32]. In general, we have inputs, outputs and a system or a function, which maps from input to output, and we want to approximate it by learning from a training set. If the output of the training set through the system or the function is known (provided by a supervisor), the learning process is called supervised learning [33].

The learning process can be seen as minimizing the training error by the least mean square (LMS), i.e.

$$F(\boldsymbol{\theta}) = \sum_i [\hat{I}(\beta_i) - I(\boldsymbol{\theta}, \beta_i)]^2, \quad (1)$$

where $\boldsymbol{\beta} = \{\beta_i\}$ is the training set, $\hat{I}(\beta_i)$ is the target output for the training sample β_i , and $I(\boldsymbol{\theta}, \beta_i)$ is the output from the hypothesized system with its parameter vector $\boldsymbol{\theta} = \{\theta_l\}$. The learning task is to train the hypothesized system to closely agree with the true mapping by continuously updating $\boldsymbol{\theta}$. In this paper, this represents the mask pattern to be optimized. The possible defocus values compose the training set $\boldsymbol{\beta}$, and the desired circuit pattern is the target output (the true mapping of the training sample). In general \hat{I} is a function of β_i . However, for our case, we will show later that \hat{I} is independent of β_i because it represents the desired mask pattern under all circumstances.

The training process amounts to minimizing an error function, which is an average over all training samples, as described in equation (1). Gradient-based searching is a preferred algorithm to solve such a problem. Batch gradient descent (offline training) updates the parameters after calculating the gradients of all samples. On the other hand,

stochastic gradient descent (online training) is typically used to fit parameters of a learning model by updating the parameters with an instant sample each time [34, 35].

For batch gradient descent (BGD), $\boldsymbol{\theta}$ is updated by

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \epsilon^{(k)} \sum_i \nabla_{\boldsymbol{\theta}} F_i(\boldsymbol{\theta}^{(k)}), \quad (2)$$

where $\nabla_{\boldsymbol{\theta}}$ is the gradient taken with respect to $\boldsymbol{\theta}$, $\epsilon^{(k)}$ is the learning rate, and $\nabla_{\boldsymbol{\theta}} F_i(\boldsymbol{\theta}^{(k)})$ is the gradient when taking a training sample β_i . In our problem, the learning rate can be replaced by a step size, which controls the amount of each update. In equation (2), every update computes gradients of all training samples. The learning process often requires the size of the training set to be large enough for precise approximation, which makes the batch gradient descent method time-consuming.

For stochastic gradient descent (SGD), however, the parameters are updated by the gradient of a single training sample each time,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \epsilon^{(k)} \nabla_{\boldsymbol{\theta}} F_i(\boldsymbol{\theta}^{(k)}). \quad (3)$$

Compared to batch gradient descent, the parameters are updated more frequently, and the convergence is faster. Fluctuation will occur due to randomness inherent in the process. We can smooth it out by setting constraints on the variance of training samples.

In this paper, we model the robust mask synthesis with focus variation as an online learning problem. By adopting the stochastic gradient descent method, the mask is continuously adapted to various focus conditions, and consequently gains its robustness. The training process involves solving an optimization problem. Therefore, in what follows, we will first present the mathematical model of inverse lithography, and then build the optimization framework to synthesize the robust optimal masks. We also compare results generated from the standard gradient descent method (GD), which optimizes masks for a nominal condition, and a previous study on mask robustness, with our method. The mask performance and algorithm efficiency are discussed in detail.

2. Optimization framework for robust mask synthesis

We use image synthesis to solve the mask design problem, and the pixel-based representation is adopted. In this paper, all images are represented by 2D matrices. We denote the original design, the mask image, and the printed on-wafer pattern by \hat{I} , M , and I respectively. For simplicity, we assume a coherent imaging system and a binary mask.

The optical lithography imaging system is illustrated in figure 1. The light source illuminates the mask, projecting it onto the image plane. The nominal image plane is located at the focal plane. However, the location of the real image plane fluctuates with respect to the distances from the main lens, resulting in defocus aberration. The distance between the real image plane and the focal plane is then the defocus denoted by β . The projected image, which is also called the aerial image,

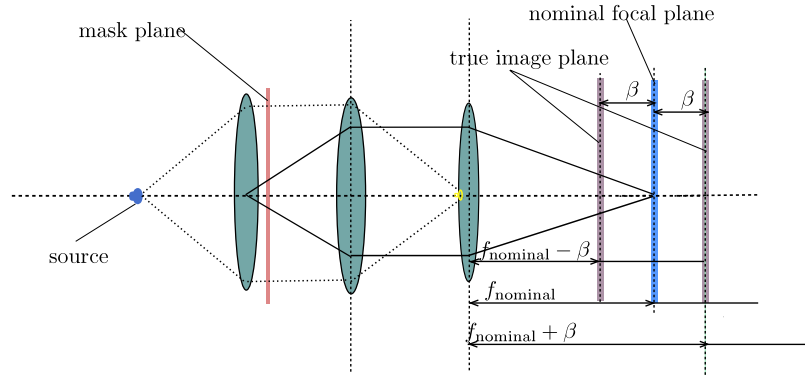


Figure 1. The defocus model in an optical projection lithography system.

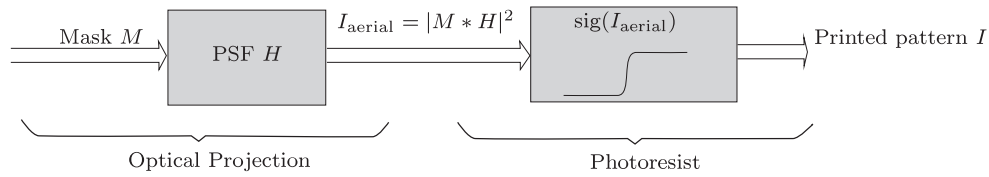


Figure 2. Forward model of the optical lithography system.

is recorded on the wafer coated by the photoresist. The exposed region of the photoresist then goes through a development procedure to retain the imaged region on the wafer. According to the above procedures, the lithography process is divided into two parts, which are modeled by the mathematical descriptions below.

The projection procedure is modeled as a mask image convolved with the point spread function (PSF), as illustrated in figure 2. For coherent imaging, the PSF is given by

$$H(x, y) = \mathcal{F}\{\tilde{H}(f, g)\}, \quad (4)$$

where

$$\tilde{H}(f, g) = \begin{cases} 1 & \text{when } \sqrt{f^2 + g^2} \leq \frac{NA}{\lambda} \\ 0 & \text{when } \sqrt{f^2 + g^2} > \frac{NA}{\lambda}. \end{cases} \quad (5)$$

Here f and g are spatial frequency variables, NA the numerical aperture, λ the wavelength, and \tilde{H} is the optical transfer function (OTF).

A lithography system with defocus is modeled as a phase shift of the OTF in the frequency domain, i.e.

$$\tilde{H}(f, g; \beta) = \tilde{H}(f, g)e^{-j\pi\beta(f^2+g^2)}, \quad (6)$$

where $j = \sqrt{-1}$. Here we obtain the PSF with defocus $H(x, y; \beta)$, which is the Fourier transform of $\tilde{H}(f, g; \beta)$. The aerial image I_A is then calculated by

$$I_A(x, y; \beta) = |M(x, y) * H(x, y; \beta)|^2, \quad (7)$$

where the symbol $*$ denotes the convolution operator. The action involving the photoresist can be modeled by a continuous sigmoid function. Thus the printed pattern I is represented by the output of a sigmoid function of the aerial image, i.e.

$$I(x, y; \beta) = \text{sig}(I_A(x, y; \beta)) = \{1 + e^{-\alpha(I_A(x, y; \beta) - t_r)}\}^{-1}, \quad (8)$$

where α defines the steepness of the sigmoid function, and t_r is the threshold above which the photoresist is developed.

Given the forward system model and the original design, the photomask design problem can be computed by solving an inverse imaging problem using optimization. Without considering process variations, a general approach is to minimize the mean square error (MSE) of the printed pattern $I(x, y; \beta = 0)$ calculated at best focus with respect to the target design $\hat{I}(x, y)$. In this paper, we aim to tackle a robust photomask design problem by introducing the defocus as a Gaussian distributed random variable. The imaging system with defocus is no longer deterministic, but stochastic. In this case, we minimize the expectation of the MSE

$$\begin{aligned} & \text{minimize } \mathcal{E}_\beta \{\|I(x, y; \beta) - \hat{I}(x, y)\|_2^2\} \\ & \text{subject to } M(x, y) \in \{0, 1\}. \end{aligned} \quad (9)$$

In equation (9), \mathcal{E}_β is the expectation operator with respect to β , and $\|\cdot\|_2$ denotes the ℓ_2 norm. Note that our work is not just restricted to chrome-on-glass (CoG) mask. The same framework can also be applied to attenuated phase-shifting mask (PSM), where $M(x, y) \in \{-0.2646, 1\}$ [1], or alternating PSMs, where $M(x, y) \in \{-1, 0, 1\}$, although the last case would make the computation more challenging.

The constraint $M(x, y) \in \{0, 1\}$ makes the above a combinatorial optimization problem. One common approach is to relax this constraint to $0 \leq M(x, y) \leq 1$. Trigonometry is used to transform this optimization problem to an unconstrained one as [11]

$$M(x, y) = \frac{1 + \cos \theta(x, y)}{2}. \quad (10)$$

Consequently, the search for an optimal $M(x, y)$ is equivalent to finding the $\theta(x, y)$ which gives the minimum of the cost

function

$$F = \mathcal{E}_\beta \left\{ \sum_{x,y} [I(x, y; \beta) - \hat{I}(x, y)]^2 \right\}. \quad (11)$$

Due to the nonlinearity of equation (9), the analytical form of the expectation is difficult to derive. So we approximate it by a discrete form

$$F = \sum_i \eta_i \left\{ \sum_{x,y} [I(x, y; \beta_i) - \hat{I}(x, y)]^2 \right\}, \quad (12)$$

where η_i represents the probability density function of β_i .

If we treat F , $\theta = \{\theta(x, y)\}$, and $\beta = \{\beta_i\}$ in equation (12) (where $I(x, y; \beta_i)$ is defined by $\theta(x, y)$) as the error function, the parameters of the learning system, and the observation samples respectively as in equation (1), the search for a robust mask M then involves training the system parameters θ to adapt to a training set β . The desired pattern $\hat{I}(x, y)$ is the target output in a machine learning problem, and the layout $I(x, y; \beta_i)$ is the hypothesized system output of a training sample β_i .

Notice that here $\theta(x, y)$ is a matrix with the same dimension as $M(x, y)$. The training is executed by iteratively modifying the parameters $\theta(x, y)$ using the training set. As mentioned in section 1.2, this is achieved through solving an optimization problem, typically using gradient-based methods. Thus the parameters are updated by the gradient of the error function F , i.e., the derivative of F with respect to θ .

Let us use \hat{I} , I , M , θ and $H(\beta_i)$ to stand for $\hat{I}(x, y)$, $I(x, y; \beta_i)$, $M(x, y)$, $\theta(x, y)$, and $H(x, y; \beta_i)$ respectively. Given the discrete cost function in equation (12), the gradient d to update parameters θ equals $\nabla_\theta F$, where (the full derivation is given in the appendix)

$$\begin{aligned} \nabla_\theta F = \frac{\partial F}{\partial \theta} = -\alpha \sum_i \eta_i \{ & H(\beta_i) \\ & * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H^*(\beta_i))] \\ & + H^*(\beta_i) \\ & * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H(\beta_i))] \} \odot \sin \theta. \end{aligned} \quad (13)$$

Here \odot denotes the pixel-wise multiplication, and H^* is the conjugate transpose of H . The parameter θ is adjusted by the weighted sum of gradients among all the samples in the training set. In theory, the summation in equation (13) requires an infinite sample of the defocus values $\{\beta_i\}$. In practice, we are limited to only a set of N such values, and η_i for $1 \leq i \leq N$ represents the normalized weights.

Equation (13) describes an application of batch gradient descent in which the parameter θ can only be updated after calculating all the gradients of the entire set. This offers a machine learning explanation to our previous study [25]. The stochastic optimization problem described in equation (11) is transformed to a deterministic one, which is the so-called offline training. However, if a large data set is required the computation for a single step update would be very time-consuming.

As an alternative, we can use the stochastic gradient descent to approximate $\partial F/\partial \theta$ more efficiently. As described

in equation (3), in every iteration step, the gradient is calculated under a single focus condition. Mathematically, the k th update is computed under β_i , a randomly generated defocus value following the distribution of the random variable β , and the gradient $d^{(k)}$ is therefore

$$\begin{aligned} d^{(k)} = \frac{\partial F(\beta_i)}{\partial \theta} = -\alpha \{ & H(\beta_i) \\ & * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M^{(k)} * H^*(\beta_i))] \\ & + H^*(\beta_i) \\ & * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M^{(k)} * H(\beta_i))] \} \odot \sin \theta^{(k)}. \end{aligned} \quad (14)$$

The stochastic gradient quantity $\partial F(\beta_i)/\partial \theta$ does not equal the deterministic quantity $\partial F/\partial \theta$ in general [36]. The gradient in equation (14) includes not only the difference between the output and the target pattern as well as the distortion due to diffraction, but also the amount of change caused by a focus error β_i . The mask is distorted continuously each time by the gradient computed under a different defocus value. Since β_i is randomly chosen according to its zero-mean Gaussian distribution, the training is dominated by defocus in a range which is centered at best focus with a large probability. The training therefore targets the on-wafer performance in the most possible defocus range, leading to a gain in robustness. For an extreme case of equation (14), where β_i distributes as a delta function around zero, the iteration reduces to the standard gradient descent method that optimizes mask patterns at best focus.

Given the form of the gradient, the parameter θ is updated by

$$\theta^{(k+1)}(x, y) = \theta^{(k)}(x, y) - \epsilon \cdot d^{(k)}(x, y), \quad (15)$$

where ϵ defines the step size, which can be a constant or adaptive; we adopt the former in this paper. The resulting mask is then

$$M^{(k+1)}(x, y) = \frac{1 + \cos \theta^{(k+1)}(x, y)}{2}. \quad (16)$$

Following the above procedures described by equations (14) and (15), θ is continuously trained to be adapted to a series of defocus samples.

3. Result

We compare the results of our stochastic gradient descent (SGD) algorithm with masks optimized by two other mask design methods, one using the standard gradient descent (GD) at the nominal focus condition [11], and one using several defocus samples to generate robust masks [25]. The latter, in machine learning, can be seen as a batch gradient descent (BGD) application. We use M_G , M_S , M_B to represent the optimized mask by GD, SGD, and BGD, and I_G , I_S , I_B for their corresponding output patterns respectively.

We choose three representative test patterns, including contacts, multiple gates, and a complex one. The optimized masks and their printed patterns at best focus and defocus are illustrated in figures 3–5. The white background represents the opaque region on the mask or the unexposed region on

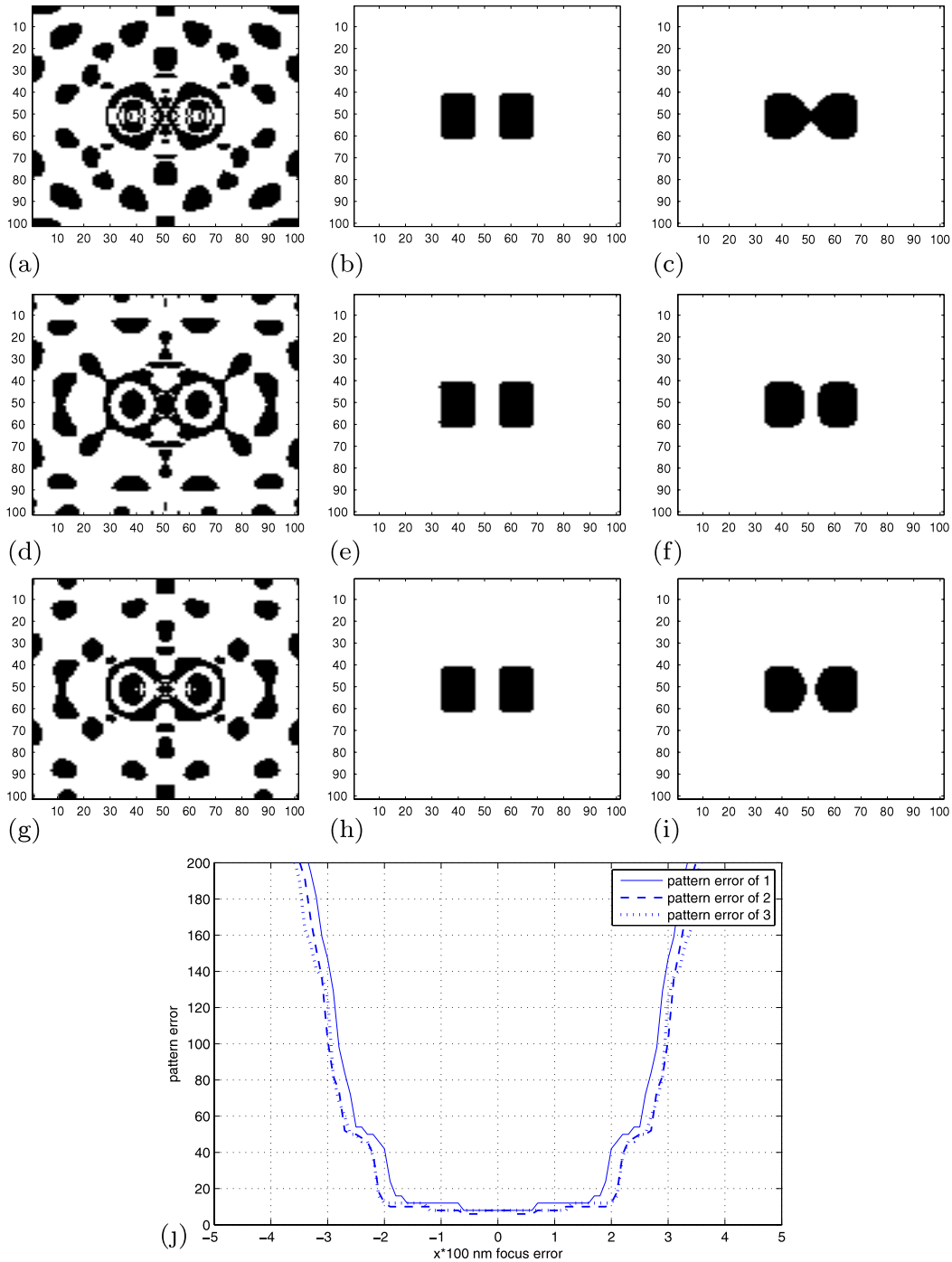


Figure 3. Results of pattern #1. Each row presents the optimized mask and its on-wafer patterns of one algorithm. From top to bottom, the three rows show results of the standard gradient descent, stochastic gradient descent, and batch gradient descent, respectively. (a) Mask M_G optimized by GD; (b) on-wafer pattern I_G at best focus, $P_e = 8$; (c) on-wafer pattern I_G at defocus 290 nm, $P_e = 129$; (d) mask M_S optimized by SGD; (e) on-wafer pattern I_S at best focus, $P_e = 8$; (f) on-wafer pattern I_S at defocus 290 nm, $P_e = 82$; (g) mask M_B optimized by BGD; (h) on-wafer pattern I_B at best focus, $P_e = 8$; (i) on-wafer pattern I_B at defocus 290 nm, $P_e = 82$ and (j) pattern #1.

the wafer, while the black shapes are the mask pattern or the printed circuit on the wafer. The robustness of the masks is assessed by computing the pattern error P_e , which evaluates the closeness between the design and the actual circuit pattern by counting the number of pixels with different values.

In this paper, we assume the focal error β follows a zero-mean Gaussian distribution with standard deviation 150 nm, and the step size $\epsilon = 2.5$.

3.1. Comparison with masks optimized at nominal conditions

The gradient descent method has been applied on inverse lithography under the best focus condition [11]. This approach can deliver on-wafer patterns close to the original design, but the performance deteriorates significantly with defocus. figure 3 illustrates the results of pattern #1, a two-rectangle mask. The three images in the first row, from left to right,

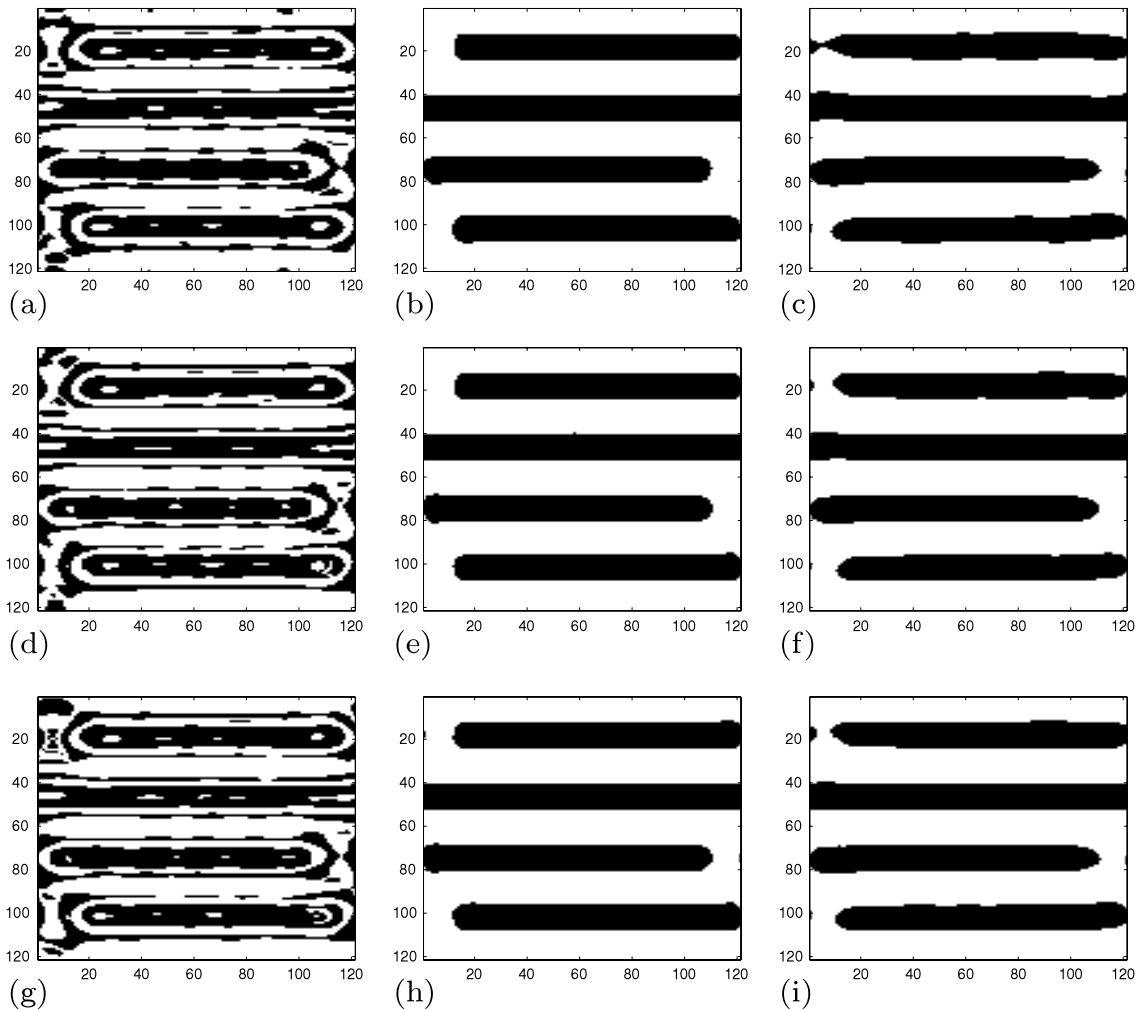


Figure 4. Results of pattern #2. The arrangement of subfigures is the same as that of figure 3. (a) Mask M_G optimized by GD; (b) on-wafer pattern I_G at best focus, $P_e = 55$; (c) on-wafer pattern I_G at defocus 360 nm, $P_e = 468$; (d) mask M_S optimized by SGD; (e) on-wafer pattern I_S at best focus, $P_e = 54$; (f) on-wafer pattern I_S at defocus 360 nm, $P_e = 223$; (g) mask M_B optimized by BGD; (h) on-wafer pattern I_B at best focus, $P_e = 67$; (i) on-wafer pattern I_B at defocus 360 nm, $P_e = 245$.

are the optimized mask by GD, its printed on-wafer patterns at best focus, and at defocus 290 nm, respectively. Following the same order, the second row shows the optimized mask using SGD and its on-wafer patterns, and the third row is for BGD.

Looking at the output patterns with no focus error, as shown in figures 3(b) and (e), we can see that GD and SGD give a similar performance of pattern fidelity. The rectangular shapes are both well printed, with the same pattern error P_e , though in general there may be slight pattern fidelity sacrifice on the mask optimized with our algorithm. When a 290 nm defocus is introduced, there is visible difference between the corresponding on-wafer patterns. The defocus bridges the two separate contacts together in (c), while in (f) the two rectangles are still distinct. Pattern failure is avoided at this defocus level by applying our algorithm. In addition, the P_e of pattern #1 is depicted in figure 3(j). The blue curve stands for the pattern error made by mask M_G in (a), while the red curve delineates the performance of mask M_S in (d). We can see that the latter admits a wider range of small pattern error than the former.

Results for the other two patterns are shown in figures 4 and 5. For the four-gate pattern in the former, the output

(c) is printed with line-end rounding and deformation. Those distortions are less in (f). In figure 5, the mask synthesized for nominal conditions suffers line width narrowing at defocus, which is shown in (c). The deviation is less serious in our output in (f). From these test figures, we see that given similar performance on the nominal printed patterns, the optimal masks generated by our algorithm show a better performance on the robustness.

3.2. Comparison with a batch gradient descent application

As mentioned in section 2, our previous investigation can be seen as an application of batch gradient descent, where we transform the stochastic problem to a deterministic one. A series of defocus values are sampled to approximate the expectation described in equation (13). Still using the above test patterns, we compare it with the work in this paper.

Let us consider figure 3 again. From left to right, the third row gives the results of the mask optimized by BGD, and its outputs at best focus and defocus. Comparing with their counterparts in the second row, which are the results generated

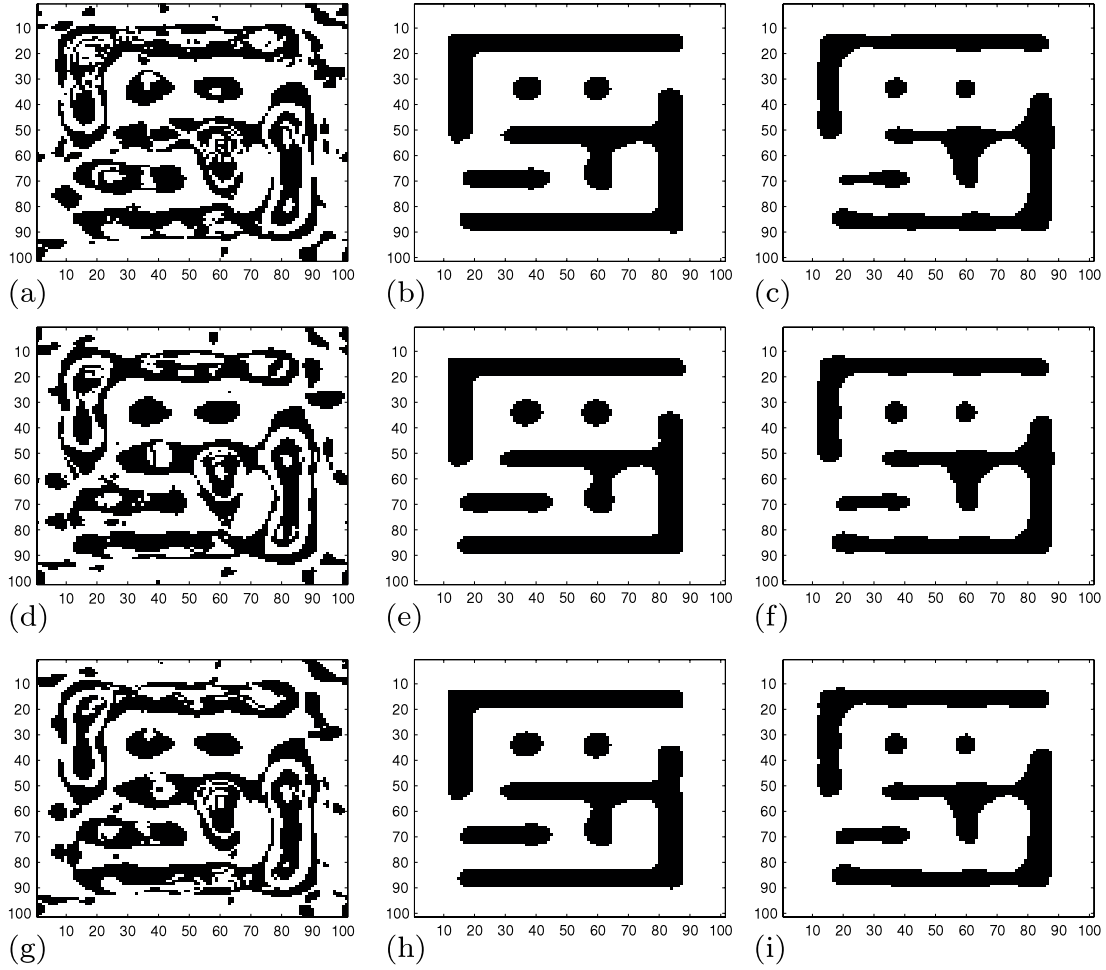


Figure 5. Results of pattern #3. The arrangement of subfigures is the same as that of figure 3. (a) Mask M_G optimized by GD; (b) on-wafer pattern I_G at best focus, $P_e = 102$; (c) on-wafer pattern I_G at defocus 350 nm, $P_e = 607$; (d) mask M_S optimized by SGD; (e) on-wafer pattern I_S at best focus, $P_e = 123$; (f) on-wafer pattern I_S at defocus 350 nm, $P_e = 456$; (g) mask M_B optimized by BGD; (h) on-wafer pattern I_B at best focus, $P_e = 104$; (i) on-wafer pattern I_B at defocus 350 nm, $P_e = 452$.

by SGD, the printed patterns (e) and (h), as well as (f) and (i), show similar performance both in terms of geometry and pattern error P_e . This similarity is further illustrated by the P_e curves in (j), where the green (BGD) and the red (SGD) ones almost overlap. Similar conclusions can be drawn from the two other patterns illustrated in figures 4 and 5.

While SGD and BGD show similar performance in terms of pattern robustness, the former has a distinct advantage in run time. Since BGD needs multiple samples to compute the gradient for one update, it costs much more computation in comparison. In our implementation, with iterations leading to similar on-wafer performances, the computation time of mask M_B is generally two to four times that of mask M_S .

4. Conclusion

This paper formulates inverse mask synthesis as a machine learning problem, and adopts the stochastic gradient descent approach to train the mask to be robust to focus variation. Experimental results show that it has comparable performance to our previous work, but requires less computation.

Acknowledgments

This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Projects HKU 7139/06E and 7134/08E, and by the UGC Areas of Excellence project *Theory, Modeling, and Simulation of Emerging Electronics*.

Appendix. Derivation of the gradient

To derive the gradient in equation (13), we first give some useful results on some partial derivatives, i.e.

$$\begin{aligned} \frac{\partial \text{sig}(x)}{\partial x} &= \frac{\partial \frac{1}{1+e^{-\alpha(x-t_r)}}}{\partial x} = \alpha \frac{1}{1+e^{-\alpha(x-t_r)}} \left(1 - \frac{1}{1+e^{-\alpha(x-t_r)}} \right) \\ &= \alpha \text{sig}(x)[1 - \text{sig}(x)], \end{aligned} \quad (\text{A.1})$$

and

$$\begin{aligned} &\frac{\partial \{M(x, y) * H(x, y; \beta_i)\}}{\partial M(p, q)} \\ &= \frac{\partial \{\sum_{m,n} M(m, n) H(x-m, y-n; \beta_i)\}}{\partial M(p, q)} \\ &= H(x-p, y-q; \beta_i). \end{aligned} \quad (\text{A.2})$$

We can express the aerial image as

$$\begin{aligned} I_A(x, y; \beta_i) &= |M(x, y) * H(x, y; \beta_i)|^2 \\ &= [M(x, y) * H(x, y; \beta_i)][M(x, y) * H(x, y; \beta_i)]^* \\ &= [M(x, y) * H(x, y; \beta_i)][M(x, y) * H^*(x, y; \beta_i)] \end{aligned} \quad (\text{A.3})$$

since $M(x, y)$ is real.

Therefore, the partial derivative of the aerial image with respect to the variable θ is

$$\begin{aligned} \frac{\partial I_A(x, y; \beta_i)}{\partial \theta(p, q)} &= \frac{[M(x, y) * H(x, y; \beta_i)][M(x, y) * H^*(x, y; \beta_i)]}{\partial M(p, q)} \\ &\times \frac{\partial M(p, q)}{\partial \theta(p, q)}, \\ &= \{[M(x, y) * H^*(x, y; \beta_i)]H(x - p, y - q; \beta_i) \\ &+ [M(x, y) * H(x, y; \beta_i)]H^*(x - p, y - q; \beta_i)\} \\ &\times \frac{-\sin \theta(p, q)}{2}, \end{aligned} \quad (\text{A.4})$$

because $M(p, q) = (1 + \cos \theta(p, q))/2$.

Since $\hat{I}(x, y)$, $I(x, y; \beta_i)$, $I_A(x, y; \beta_i)$, $H(x, y; \beta_i)$, and $M(x, y)$ are matrices of many variables, we use their short forms \hat{I} , I , I_A , H and M for convenience in the following derivations. Given the cost function

$$F = \sum_i \eta_i \left\{ \sum_{x,y} (I - \hat{I})^2 \right\}, \quad (\text{A.5})$$

and the above derivations (equations (A.1)–(A.4)), the gradient ∇F is therefore

$$\begin{aligned} \frac{\partial F}{\partial \theta(p, q)} &= \sum_i \eta_i \left\{ \sum_{x,y} 2(I - \hat{I}) \frac{\partial I}{\partial \theta} \right\} \\ &= \sum_i \eta_i \left\{ \sum_{x,y} 2(I - \hat{I}) \frac{\partial \text{sig}(I_A)}{\partial \theta} \right\} \\ &= \sum_i \eta_i \left\{ \sum_{x,y} 2\alpha(I - \hat{I}) \text{sig}(I_A)[1 - \text{sig}(I_A)] \frac{\partial I_A}{\partial \theta(p, q)} \right\} \\ &= \sum_i \eta_i \left\{ \sum_{x,y} -2\alpha(I - \hat{I}) \text{sig}(I_A)[1 - \text{sig}(I_A)][(M * H^*) \right. \\ &\quad \times H(x - p, y - q; \beta_i) \\ &\quad \left. + (M * H)H^*(x - p, y - q; \beta_i)] \frac{\sin \theta(p, q)}{2} \right\} \\ &= -\alpha \sum_i \eta_i \{ H(\beta_i) \\ &\quad * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H^*(\beta_i))] \\ &\quad + H^*(\beta_i) \\ &\quad * [(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H(\beta_i))] \} \odot \sin \theta, \end{aligned} \quad (\text{A.6})$$

which is the expression in equation (13).

References

[1] Wong A K 2001 *Resolution Enhancement Techniques in Optical Lithography* (Washington: SPIE Optical Engineering Press)

[2] Liu Y and Zakhor A 1990 Optimal binary image design for optical lithography *Optical/Laser Microlithography III, Proc. SPIE* **1264** 401–12

[3] Liu Y and Zakhor A 1992 Binary and phase shifting mask design for optical lithography *IEEE Trans. Semicond. Manuf.* **5** 138–52

[4] Sherif S, Saleh B and De Leone R 1995 Binary image synthesis using mixed linear integer programming *IEEE Trans. Image Process.* **4** 1252–7

[5] Pati Y C and Kailath T 1994 Phase-shifting masks for microlithography: automated design and mask requirements *J. Opt. Soc. Am. A* **11** 2438–52

[6] Liu Y, Abrams D, Pang L and Moore A 2005 Inverse lithography technology principles in practice: unintuitive patterns *25th Annual BACUS Symp. on Photomask Technology, Proc. SPIE* **5992** 886–93

[7] Granik Y 2005 Solving inverse problems of optical microlithography *Image and Process Modeling II, Proc. SPIE* **5754** 506–26

[8] Erdmann A, Fühner T, Schnattinger T and Tollkühn B 2004 Toward automatic mask and source optimization for optical lithography *Optical Microlithography XVII, Proc. SPIE* **5377** 646–57

[9] Pang L, Liu Y and Abrams D 2007 Inverse lithography technology (ilt): a natural solution for model-based SRAF at 45 and 32 nm *Photomask and Next-Generation Lithography Mask Technology XIV, Proc. SPIE* **6607** 660739

[10] Shen Y, Wong N and Lam E Y 2009 Level-set-based inverse lithography for photomask synthesis *Opt. Express* **17** 23690–701

[11] Poonawala A and Milanfar P 2007 Mask design for optical microlithography—an inverse imaging problem *IEEE Trans. Image Process.* **16** 774–88

[12] Chan S H, Wong A K and Lam E Y 2008 Initialization for robust inverse synthesis of phase-shifting masks in optical projection lithography *Opt. Express* **16** 14746–60

[13] Poonawala A and Milanfar P 2007 Double-exposure mask synthesis using inverse lithography *J. Micro/Nanolith. MEMS MOEMS* **6** 043001

[14] Poonawala A, Painter B and Mayhew J 2008 Model-based assist feature placement: an inverse imaging approach *Photomask Technology 2008, Proc. SPIE* **7122** 71220U

[15] Ma X and Arce G R 2007 Generalized inverse lithography methods for phase-shifting mask design *Opt. Express* **15** 15066–79

[16] Chan S H and Lam E Y 2008 Inverse image problem of designing phase shifting masks in optical lithography *Proc. of IEEE Int. Conf. on Image Processing* pp 1832–5

[17] Zhang J, Xiong W, Wang Y, Yu Z and Tsai M-C 2008 A highly efficient optimization algorithm for pixel manipulation in inverse lithography technique *Proc. 2008 IEEE/ACM Int. Conf. on Computer-Aided Design* pp 480–7

[18] Jia N, Wong A K and Lam E Y 2009 Regularization of inverse photomask synthesis to enhance manufacturability *Lithography Asia 2009, Proc. SPIE* **7520** 752032

[19] Lam E Y and Wong A K 2009 Computation lithography: virtual reality and virtual virtuality *Opt. Express* **17** 12259–68

[20] Sturtevant J L, Torres J A, Word J, Granik Y and LaCour P 2005 Considerations for the use of defocus models for OPC *Design and Process Integration for Microelectronic Manufacturing III, Proc. SPIE* **5756** 427–36

[21] Cobb N B and Granik Y 2003 OPC methods to improve image slope and process window *Design and Process Integration for Microelectronic Manufacturing, Proc. SPIE* **5042** 116–25

[22] Qian Q-D and Takase S 2003 Focus latitude optimization for model-based OPC *23rd Annual BACUS Symp. on Photomask Technology, Proc. SPIE* **5256** 230–7

- [23] Zhang Q, Croffie E, Fan Y, Li J, Lucas K, Falch B and Melvin L 2009 Process variation aware OPC modeling for leading edge technology nodes *Design for Manufacturability through Design-Process Integration III, Proc. SPIE* **7275** 72751J
- [24] Yu P, Shi S X and Pan D Z 2007 True process variation aware optical proximity correction with variational lithography modeling and model calibration *J. Micro/Nanolith. MEMS MOEMS* **6** 031004
- [25] Jia N, Wong A K and Lam E Y 2008 Robust mask design with defocus variation using inverse synthesis *Lithography Asia 2008, Proc. SPIE* **7140** 71401W
- [26] Mitchell T M 1997 *Machine Learning* (New York: McGraw-Hill)
- [27] Ding D, Wu X, Ghosh J and Pan D Z 2009 Machine learning based lithographic hotspot detection with critical-feature extraction and classification *Proc. IEEE Int. Conf. on IC Design and Technology* pp 219–22
- [28] Drmanac D G, Liu F and Wang L-C 2009 Predicting variability in nanoscale lithography processes *Proc. 46th Annual IEEE Design Automation Conf. (San Francisco)* pp 545–50
- [29] Frye R C, Rietman E A and Cummings K D 1990 Neural network proximity effect corrections for electron beam lithography *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics* pp 704–6
- [30] Jedrasik P 1996 Neural networks application for OPC (optical proximity correction) in mask making *Microelectron. Eng.* **30** 161–4
- [31] Gu A and Zakhor A 2008 Optical proximity correction with linear regression *IEEE Trans. Semicond. Manuf.* **21** 263–71
- [32] Nilsson N J 1996 Introduction to machine learning <http://robotics.stanford.edu/people/nilsson/mlbook.html>
- [33] Alpaydin E 2004 *Introduction to Machine Learning* (Cambridge, MA: MIT Press)
- [34] Bottou L 1991 Stochastic gradient learning in neural networks *Proc. Neuro-Nimes* vol 91, pp 687–696
- [35] Plagianakos V P, Magoulas G D and Vrahatis M N 2001 Learning rate adaptation in stochastic gradient descent *Nonconvex Optimization and its Application* vol 54 (Dordrecht: Kluwer) pp 433–44
- [36] Spall J C 2003 *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control* (Hoboken, NJ: Wiley)