STOCHASTIC GRADIENT DESCENT FOR ROBUST INVERSE PHOTOMASK SYNTHESIS IN OPTICAL LITHOGRAPHY

Ningning Jia and Edmund Y. Lam

Imaging Systems Laboratory, Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong.
{nnjia,elam}@eee.hku.hk

ABSTRACT

Optical lithography is a critical step in the semiconductor manufacturing process, and one key problem is the design of the photomask for a particular circuit pattern, given the optical aberrations and diffraction effects associated with the small feature size. Inverse lithography synthesizes an optimal mask by treating the design as an image synthesis inverse problem. To date, much effort is dedicated to solving it for some nominal process conditions. However, the small feature size also suggests that the effect of process variations is more pronounced. In this paper, we design a mask that is robust against focus variations within the inverse lithography framework. Each iteration involves more computation than a similar method designed for the nominal conditions, but we simplify the task by using stochastic gradient descent, which is a technique from machine learning. Simulation shows that the proposed algorithm is effective in producing robust masks.

***Index Terms*—** Inverse imaging, lithography, optical proximity correction, robustness, stochastic gradient descent, machine learning

1. INTRODUCTION

The rapid advancement of the electronic industry is driven, in large part, by the continuation of Moore's Law. This dictates that the transistors have to packed more and more closely in an integrated circuit; in other words, the circuit feature size has to be made smaller and smaller. This poses important challenges to the semiconductor manufacturing process, particularly in optical lithography.

Consequently, many resolution enhancement techniques have been invented for lithography. One is called optical proximity correction (OPC), which pre-distorts the mask pattern in view of the distortions and diffraction effects in the imaging process. Inverse lithography technology (ILT) is an emerging approach to OPC, which aims to synthesize the mask by solving an inverse imaging problem. It manipulates mask pixels by optimizing an appropriate function. Since ILT was first proposed in 1990s [1], many algorithms have been developed [2, 3, 4]. However, two important concerns remain: one is that the resulting mask needs to be regularized for manufacturability, and second is that the mask needs to be robust against process variations, which is an increasingly important criterion given the lower k_1 factor in the imaging system [5]. In this paper, we tackle the latter by incorporating focus variation in the problem formulation, and solving each iteration using stochastic gradient descent from machine learning.

Machine learning is concerned with constructing programs that automatically improve its behavior with experience [6]. Generally, we have inputs, outputs and a system or a function, which maps input to output, and we want to approximate it by learning from a training set [7]. The learning process can be seen as minimizing the training error by the least mean square (LMS), i.e.

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

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 comprise the training set $\boldsymbol{\beta}$, and the desired circuit pattern is the target output (the true mapping). 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 Eq. 1. Gradient-based searching, such as batch gradient descent and stochastic gradient descent, is a preferred algorithm to solve such a problem [8].

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

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

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

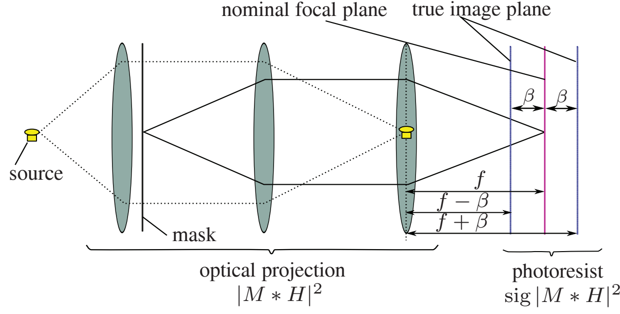


Fig. 1: The defocus model in an optical projection lithography system.

training sample β_i. In our case, the learning rate can be replaced by a step size, which controls the amount of each update. In Eq. 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 BGD method time-consuming.

On the other hand, one can use the stochastic gradient descent (SGD), where the parameters are updated by the gradient of a single training sample each time, i.e.

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

Compared to BGD, 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 by setting constraints on the variance of the training samples.

In the following, we will first develop the optimization framework for inverse robust mask synthesis, and solve it as a training problem using stochastic gradient descent. Results are compared with the other two mask design approaches, namely using gradient descent for the nominal process condition, and batch gradient descent.

2. OPTIMIZATION FRAMEWORK

This paper adopts the pixel-based representation, hence all images are represented by 2-D 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 Fig. 1, with a detailed description of the system model in [9]. Given a mask pattern, the printed image is calculated by

$$I(x,y;\beta) = \left[1 + e^{-\alpha(|M(x,y)*H(x,y;\beta)|-t_r)}\right]^{-1}, \tag{4}$$

where α defines the steepness of the sigmoid function, t_r is the threshold, and H denotes the point spread function (PSF). This model, in particular the use of the sigmoid function to represent the resist process, is consistent with other work on inverse lithography [10, 11].

Given the forward system model and the original design, the photomask 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 [9]. The imaging system with defocus is no longer deterministic but stochastic. In this case, we minimize the expectation of the MSE

$$\begin{aligned} \text{minimize} \quad & F = \mathcal{E}_\beta \left\{ \sum_{x,y} \left[I(x,y;\beta) - \hat{I}(x,y) \right]^2 \right\}. \\ \text{subject to} \quad & M(x,y) \in \{0,1\}, \end{aligned} \tag{5}$$

where $\mathcal{E}_\beta$ is the expectation operator with respect to β. The constraint $M(x,y) \in \{0,1\}$ makes the above a combinatorial optimization problem. One common approach is to relax this constraint to $0 \leq M(x,y) \leq 1$ [10]. Trigonometry is used to transform this optimization problem to an unconstrained one as [4]

$$M(x,y) = \frac{1}{2}\left[1 + \cos\theta(x,y)\right]. \tag{6}$$

Consequently, the search for an optimal $M(x,y)$ is equivalent to finding $\theta(x,y)$, which minimizes the cost function F.

Due to the nonlinearity of Eq. 5, we do not aim to derive the analytical form of the expectation, but approximate it by a discrete form

$$F = \sum_i \eta_i \left\{ \sum_{x,y} \left[I(x,y;\beta_i) - \hat{I}(x,y) \right]^2 \right\}, \tag{7}$$

where η_i represents the probability density function of β_i. Note that $I(x,y;\beta_i)$ is a function of β_i and $M(x,y)$, which is in turn defined by $\theta(x,y)$.

If we treat F, $\boldsymbol{\theta} = \{\theta(x,y)\}$, and $\boldsymbol{\beta} = \{\beta_i\}$ in Eq. 7 as the error function, the parameters of the learning system, and the observation samples respectively as in Eq. 1, the search for a robust mask M then involves training the system parameters $\boldsymbol{\theta}$ to adapt to a training set $\boldsymbol{\beta}$. The desired pattern $\hat{I}(x,y)$ is the target output, 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 earlier, 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 $\boldsymbol{\theta}$.

Let us use $\hat{I}$, I, M, θ and H_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 Eq. 7, the gradient d to update the parameters θ is equal to $\nabla_\theta F$, where

$$\nabla_\theta F = \frac{\partial F}{\partial \theta} = -\alpha \sum_i \eta_i \left\{ H_i * \left[(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H_i^*) \right] + H_i^* * \left[(I - \hat{I}) \odot I \odot (1 - I) \odot (M * H_i) \right] \right\} \odot \sin\theta. \quad (8)$$

Here $\odot$ denotes the pixel-wise multiplication, and H^* is the conjugate transpose of H. The derivation of Eq. 8 is similar to that in [4]. The parameter θ is adjusted by the weighted sum of gradients among all the samples in the training set. In theory, the summation in Eq. 8 requires an infinite sample of the defocus values $\{\beta_i\}$. In practice, we limit to only a set of N such values, and η_i for $1 \leq i \leq N$ represents the normalized weights.

Eq. 8 describes an application of batch gradient descent [9], in which the parameter θ can only be updated after calculating all the gradients. The stochastic optimization problem described in Eq. 5 is transformed to a deterministic one. However, if a large data set is involved, 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 [12]. As described in Eq. 3, in every iteration step, the gradient is calculated under a single focus condition. Mathematically, the kth 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

$$d^{(k)} = \frac{\partial F(\beta_i)}{\partial \theta} = -\alpha \left\{ H_i * \left[(I - \hat{I}) \odot I \odot (1 - I) \odot (M^{(k)} * H_i^*) \right] + H_i^* * \left[(I - \hat{I}) \odot I \odot (1 - I) \odot (M^{(k)} * H_i) \right] \right\} \odot \sin\theta^{(k)}. \quad (9)$$

The gradient in Eq. 9 includes not only the difference between the output and the target pattern and the distortion due to diffraction, but also the amount of change caused by a focus error β_i. The mask is distorted continuously by the gradient computed under a different defocus value each time. Since β_i is randomly chosen according to its zero-mean Gaussian distribution, the training is dominated by a defocus range centered at best focus with a large probability. The training therefore targets at the on-wafer performance in the most possible defocus range, through which robustness is attained. For an extreme case of Eq. 9 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, θ is updated by

$$\theta^{(k+1)}(x, y) = \theta^{(k)}(x, y) - \epsilon \cdot d^{(k)}(x, y), \quad (10)$$

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

$$M^{(k+1)}(x, y) = \frac{1}{2} \left[1 + \cos\theta^{(k+1)}(x, y) \right]. \quad (11)$$

Following the above procedures described by Eq. 9 and 10, θ is continuously trained to be adapted to a series of defocus samples.

3. RESULTS

We compare the results of our SGD algorithm with masks optimized by two other mask design methods, one using the standard gradient descent (GD) at the nominal focus condition [4], and one using BGD to generate robust masks [9]. We use M_{GD}, M_{SGD}, M_{BGD} to represent the optimized mask by GD, SGD, and BGD, and I_{GD}, I_{SGD}, I_{BGD} for their corresponding output patterns respectively.

The optimized masks and their printed patterns at best focus and defocus of a mask pattern are illustrated in Fig. 2. The white background represents the opaque region on the mask or the unexposed region on 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 150nm, and the step size $\epsilon = 2.5$.

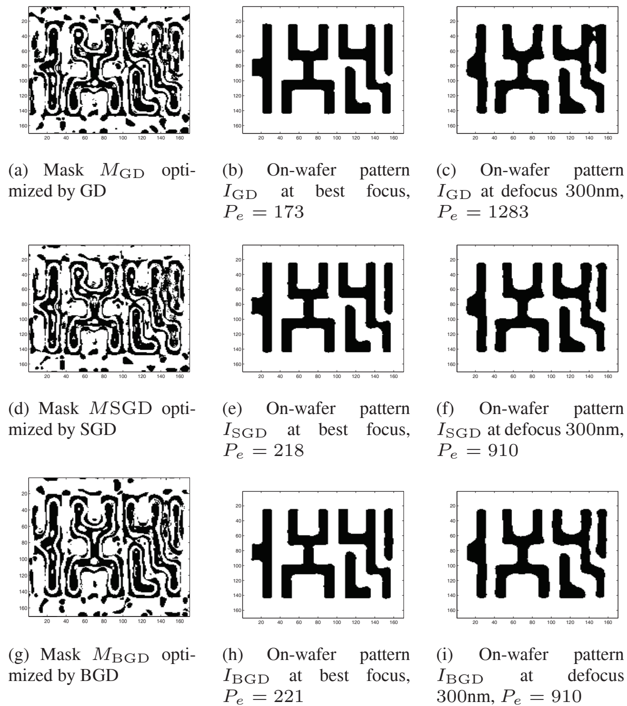
3.1. Comparison with masks optimized at nominal conditions

The gradient descent method has been applied on inverse lithography under the best focus condition [4, 11]. This approach can deliver on-wafer patterns close to the original design, but the performance deteriorates significantly with defocus. Fig. 2 illustrates the results of a test pattern.

By looking at the output patterns with no focus error, as shown in Fig. 2(b) and (e), we can see that GD and SGD give similar performance of pattern fidelity. The feature shapes are both well printed, though there is slight pattern fidelity sacrifice on the mask optimized with our algorithm. When a 300nm defocus is introduced, there is visible difference between the corresponding on-wafer patterns. The defocus bridges two adjacent features together in (c), while in (f) they are still distinct. Pattern failure is avoided at this defocus level by applying our algorithm.

3.2. Comparison with the batch gradient descent method

The batch gradient descent method has been applied to solve the robust mask design problem in our previous work [9], where a series of defocus values are sampled to approximate



(a) Mask M_{GD} optimized by GD
(b) On-wafer pattern I_{GD} at best focus, $P_e = 173$
(c) On-wafer pattern I_{GD} at defocus 300nm, $P_e = 1283$
(d) Mask $M\text{SGD}$ optimized by SGD
(e) On-wafer pattern I_{SGD} at best focus, $P_e = 218$
(f) On-wafer pattern I_{SGD} at defocus 300nm, $P_e = 910$
(g) Mask M_{BGD} optimized by BGD
(h) On-wafer pattern I_{BGD} at best focus, $P_e = 221$
(i) On-wafer pattern I_{BGD} at defocus 300nm, $P_e = 910$

Fig. 2: Results of the test pattern. 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.

the expectation described in Eq. 8. We compare it with the work in this paper.

Let us consider Fig. 2 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 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.

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_{BGD} is about four times that of mask M_{SGD}.

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 with our previous work that employs a batch gradient descent scheme, but requires less computation.

5. REFERENCES

[1] Yong Liu and Avideh Zakhor, “Binary and phase shifting mask design for optical lithography,” *IEEE Trans. Semicond. Manuf.*, vol. 5, no. 2, pp. 138–152, May 1992.

[2] Yuri Granik, “Solving inverse problems of optical microlithography,” in *Optical Microlithography XVIII*, 2005, vol. 5754 of *Proc. SPIE*, pp. 506–526.

[3] Yijiang Shen, Ngai Wong, and Edmund Y. Lam, “Level-set-based inverse lithography for photomask synthesis,” *Opt. Express*, vol. 17, no. 26, pp. 23690–23701, Dec 2009.

[4] Amyn Poonawala and Peyman Milanfar, “Mask design for optical microlithography — an inverse imaging problem,” *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 774–788, Mar 2007.

[5] Edmund Y. Lam and Alfred K. Wong, “Computation lithography: virtual reality and virtual virtuality,” *Opt. Express*, vol. 17, no. 15, pp. 12259–12268, Jul 2009.

[6] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, New York, 1997.

[7] Nils J. Nilsson, “Introduction to machine learning,” http://robotics.stanford.edu/people/nilsson/mlbook.html, 1996.

[8] Láon Bottou, “Stochastic gradient learning in neural networks,” in *Proceedings of Neuro-Nimes*, 1991, vol. 91, pp. 687–696.

[9] Ningning Jia, Alfred K. Wong, and Edmund Y. Lam, “Robust mask design with defocus variation using inverse synthesis,” in *Lithography Asia 2008*, 2008, vol. 7140 of *Proc. SPIE*, p. 71401W.

[10] Amyn Poonawala and Peyman Milanfar, “Prewarping techniques in imaging: applications in nanotechnology and biotechnology,” in *Computational Imaging III*, 2005, vol. 5674 of *Proc. SPIE*, pp. 114–127.

[11] Stanley H. Chan, Alfred K. Wong, and Edmund Y. Lam, “Initialization for robust inverse synthesis of phase-shifting masks in optical projection lithography,” *Opt. Express*, vol. 16, no. 19, pp. 14746–14760, Sep 2008.

[12] Ningning Jia and Edmund Y. Lam, “Machine learning for inverse lithography: Using stochastic gradient descent for robust photomask synthesis,” *J. Opt.*, vol. 12, no. 4, pp. 045601, 2010.