Journal of
Micro/Nanolithography,
MEMS, and MOEMS

Nanolithography.SPIEDigitalLibrary.org

Optical proximity correction with hierarchical Bayes model

Tetsuaki Matsunawa
Bei Yu
David Z. Pan

Tetsuaki Matsunawa, Bei Yu, David Z. Pan, "Optical proximity correction with hierarchical Bayes model," *J. Micro/Nanolith. MEMS MOEMS* **15**(2), 021009 (2016), doi: 10.1117/1.JMM.15.2.021009.

Optical proximity correction with hierarchical Bayes model

Tetsuaki Matsunawa,[a,*] Bei Yu,[b] and David Z. Pan[c]
[a]Toshiba Corporation, Yokohama 247-8585, Japan
[b]The Chinese University of Hong Kong, CSE Department, NT, Hong Kong
[c]The University of Texas at Austin, ECE Department, Austin, Texas 78712, United States

Abstract. Optical proximity correction (OPC) is one of the most important techniques in today's optical lithography-based manufacturing process. Although the most widely used model-based OPC is expected to achieve highly accurate correction, it is also known to be extremely time-consuming. This paper proposes a regression model for OPC using a hierarchical Bayes model (HBM). The goal of the regression model is to reduce the number of iterations in model-based OPC. Our approach utilizes a Bayes inference technique to learn the optimal parameters from given data. All parameters are estimated by the Markov Chain Monte Carlo method. Experimental results show that utilizing HBM can achieve a better solution than other conventional models, e.g., linear regression-based model, or nonlinear regression-based model. In addition, our regression results can be used as the starting point of conventional model-based OPC, through which we are able to overcome the runtime bottleneck. *© 2016 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JMM.15.2.021009]

Keywords: lithography; optical proximity correction; hierarchical Bayes model; machine learning; model-based optical proximity correction.

Paper 15162SSP received Oct. 26, 2015; accepted for publication Feb. 16, 2016; published online Mar. 11, 2016.

1 Introduction

Although several types of emerging lithography techniques, e.g., extremely ultraviolet lithography,[1] electron beam lithography,[2] nano-imprint lithography (NIL),[3] and directed self-assembly lithography,[4] are being developed, optical lithography is still widely used in the semiconductor industry in view of its cost.[5] In the optical lithography-based manufacturing process, optical proximity correction (OPC) is one of the most important techniques.[6] Figure 1 shows an overview of the most widely used model-based OPC in which the displacements of fragmented edges in a mask layout are calculated based on a lithography simulation so that the difference between target edge and simulated image can be minimized.[7] Although this method is expected to achieve highly accurate correction, it is also known to be extremely time-consuming. To resolve this issue, several fast mask optimization methods, such as inverse lithography, linear regression-based OPC, and nonlinear regression-based OPC, have been proposed.[8–11]

The inverse lithography technique is expected to obtain a highly accurate prediction model because the ideal mask shape is derived from inverse transformation of the desired resist image on a wafer. However, this method cannot be applied to full-chip calculation owing to huge computational cost.[8] Meanwhile, from the viewpoint of runtime, regression-based methods can be promising candidates to reduce the runtime of model-based OPC, because the displacements of fragmented edges in a layout are quickly estimated through a comparatively simple regression model, which is trained with supervised displacement data calculated by the conventional model-based OPC technique. Since these methods use a straightforward regression technique, they are capable to be applied to full-chip calculation. Conventional regression-based methods, however, have several issues in terms of prediction accuracy.

A linear regression-based OPC method is proposed by Gu and Zakhor[9] and it showed the capability of reducing OPC runtime by using the regression results as a start point of model-based OPC. Also, nonlinear regression-based OPC methods using support vector machine or artificial neural network (ANN) are presented[10,11] and showed the possibility of runtime reduction against model-based OPC. However, it is challenging to train a robust regression model owing to the over-fitting problem, whereby a model has poor predictive performance. Furthermore, as device feature sizes continue shrinking, it is increasingly difficult to achieve a highly accurate prediction model owing to the model complexity, since the correction amounts for edge displacements greatly vary in accordance with the surrounding environment of fragmented edges and optical proximity effects (OPEs). In order to overcome these problems that inhibit the practical use of regression-based methods, this paper proposes a new regression-based OPC framework. We develop a hierarchical Bayes model (HBM) to achieve a robust prediction model while preventing the over-fitting issue. Moreover, we propose a new layout representation technique to improve the prediction accuracy.

The remainder of this paper is organized as follows: Sec. 2 presents the background of this work by referring to conventional regression-based OPC approaches. Section 3 gives the problem formulation and the overall flow of the proposed method. Section 4 discusses the proposed layout feature extraction technique. Section 5 describes the HBM

*Address all correspondence to: Tetsuaki Matsunawa, E-mail: tetsuaki.matsunawa@toshiba.co.jp

1932-5150/2016/$25.00 © 2016 SPIE

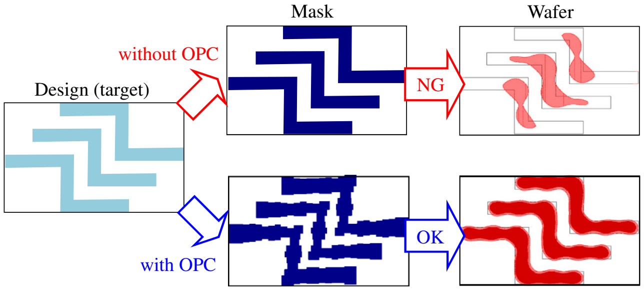


Fig. 1 Optical proximity correction.

training method. Section 6 presents the experimental results, followed by the conclusion in Section 7.

2 Preliminaries

Several regression-based OPC methods have been proposed and shown to be effective for reducing the runtime of model-based OPC. However, as layout features become complicated in the future technology nodes, it is difficult to learn a highly accurate prediction model using straightforward linear regression technique. Specifically, in the linear regression-based OPC,[9] the following simple linear model involves a linear combination of the input layout feature vectors.

$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1x_1 + \cdots + w_Dx_D, \tag{1}$$

where D is the number of dimensions of feature vectors $\mathbf{x}$ described as $\mathbf{x} = (x_1, \dots, x_D)^{\mathrm{T}}$, $\mathbf{w} = (w_0, \dots, w_D)^{\mathrm{T}}$, and w_0 is the bias parameter. The details of the feature are described in Sec. 5. The coefficients of this model $\mathbf{w}$ can be written by using the following normal equations in which $\mathbf{w}$ are derived so that the difference between a predicted edge displacement and a supervised data is minimized.

$$\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}, \tag{2}$$

where $\mathbf{X}$ is the design matrix, whose elements are given by $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^{\mathrm{T}}$, N is the total number of supervised data samples, and $\mathbf{y} = (y_1, \dots, y_N)^{\mathrm{T}}$ are the target vectors corresponding to the edge displacements obtained by model-based OPC. Although the least square method is applicable to uncomplicated regression problems, it includes the following two disadvantages: (1) Over-fitting: model learning with limited samples or nonrepresentative samples causes over-fitting or overgeneralization that lowers predictive performance of the regression model. (2) Limited applications: it is difficult to apply the linear regression method to complex nonlinear phenomena because the algorithm includes a linear assumption in which all model parameters are linearly correlated with input feature vectors.

To model complex nonlinear problems, several nonlinear regression techniques based on ANN or support vector regression (SVR) have been proposed.[10,11] These related works showed that it is possible to model nonlinear problems through mechanisms of interconnected neurons[10] or kernel methods.[11] However, the over-fitting issue is still open since a sufficient amount of supervised data is not necessarily given in all cases. Furthermore, practical applications of the regression model are also restricted even with nonlinear algorithms because there are relatively few adjustable model parameters that can improve predictive performance.

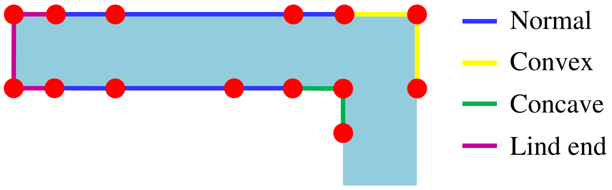


Fig. 2 Examples of edge types.

On the one hand, OPEs from adjacent patterns contribute to one of the factors that increases difficulty of model training. This is because the tendency of displacement of mask edges differs among different types of edges.[9] Figure 2 shows examples of edge types including normal edge, convex edge, concave edge, or line end edge. The difference of edge types leads to different displacement amounts. In the linear regression-based OPC,[9] an accurate regression model is realized by separately learning a regression model per edge type. Since the separate model requires a larger number of supervised data, this approach also faces the over-fitting issue. A new efficient regression algorithm is required to realize flexible modeling for complex phenomena consisting of a small number of data.

3 Problem Formulation and Overall Flow

3.1 *Problem Formulation*

To evaluate the performance of regression-based OPC, we define the RMSE as follows:

Definition 1 (root mean square error: RMSE).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_i^N (y_i - \hat{y}_i)^2}{N}}, \tag{3}$$

where N is the total number of supervised data samples, y_i is the fragment movement on i'th edge determined by model-based OPC and $\hat{y}$ is the predicted fragment movement.

We give the problem formulation of regression-based OPC as follows:

Problem 1 (regression-based OPC). Given layout data including the displacement amount of all fragments, a regression model is calibrated to predict displacements of unknown fragments in a verification layout. The goal of the regression-based OPC is to minimize the RMSE.

3.2 *Overall Flow*

Our regression-based OPC method consists of two phases, a learning phase and a testing phase as shown in Fig. 3. In the beginning of the learning phase, a training layout data is given.

Then based on a design rule check (DRC), all the edges of the training layout are fragmented into different types, such as normal edge, convex edge, concave edge, and line end edge (see Fig. 2). After recognizing these different types, the displacement amount of each fragment is computed by using the conventional model-based OPC technique as

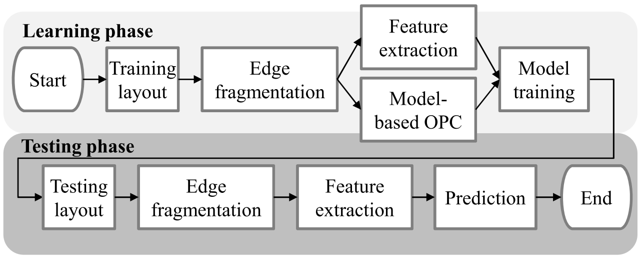


Fig. 3 Overview of the proposed method.

a supervised data. Meanwhile, layout features in each fragmented edge are extracted as described in Sec. 5. Finally, a prediction model is trained using the supervised displacement data obtained by model-based OPC and the extracted layout features.

In the testing phase, a verification layout data is used as an input. Based on the DRC in the learning phase, all edges of the layout are fragmented and edge types are recognized as well. After layout feature extraction, the edge displacements in the verification layout are predicted by the model trained in the learning phase.

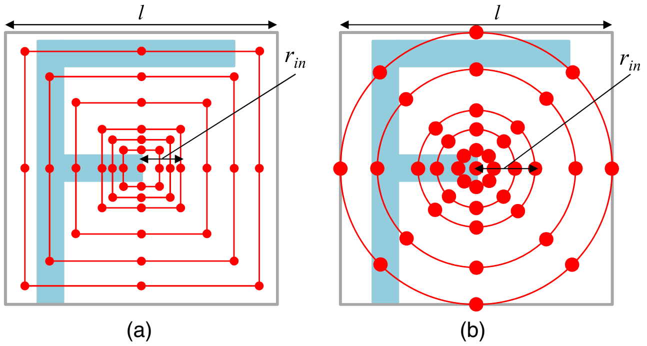
4 Feature Extraction

In this section, we discuss a layout feature extraction method for OPC regression. In a prediction model, the input layout is difficult to be directly handled, owing to its high-dimensional space. Therefore, the geometrical information of each fragmented edge is encoded into a feature vector. For instance, assuming there are 0.6 μm^2 areas on 1 nm grid to consider edge displacement of a pattern, the number of dimensions of the pattern becomes 600×600. In this kind of high-dimensional space, $\mathbb{R}^{360000}$, it is difficult to prepare a sufficient number of supervised samples and solve the regression problem within a practical time.

Layout feature extraction is a very important procedure in the regression-based OPC because the prediction model performance is heavily determined by the types of layout features. Recently, concentric square sampling (CSS)[9] has been proposed as an OPC modeling feature and showed reasonable prediction accuracy. In this paper, we propose a new layout feature extraction method, concentric circle area sampling (CCAS), for further improvement of OPC regression. In the following, we first present an overview of the CCS and then our proposed CCAS is introduced.

4.1 *Concentric Square Sampling*

CSS[9] is proposed in the linear regression-based method to train a linear model for OPC regression. A feature vector $\mathbf{x}$ contains subsampled pixel values on concentric squares of layout patterns. Figure 4(a) shows the basic concept of CSS of F-shaped test pattern. Parameters of a feature consist of the total size of the encoding area l and the sampling density controlling parameter r_{in}. The radius of the concentric square is $0, 4, 8, \cdots, r_{\text{in}}, r_{\text{in}} + 8, r_{\text{in}} + 16, \cdots, l/2$ pixels, respectively. The total number of dimensions will be 257 if the parameters l and r_{in} are 0.4 μm and 60 nm, respectively. It can be expected to achieve a high generalization capability because the feature can correctly express a positional relationship to layout patterns. Although CSS can reduce the number of dimensions compared to exploring all pixel values in the layout patterns, prediction model training might remain difficult because the number of dimensions is still high.



(a)

(b)

Fig. 4 Layout features: (a) CSS and (b) CCAS.

4.2 *Concentric Circle Area Sampling*

This feature is first presented in this paper and represents pattern information that affects propagation of diffracted light from a mask pattern. Figure 4(b) shows the basic concept of CCAS of F-shaped test pattern. Parameters of a feature consist of the total size of the encoding area l and the sampling density controlling parameter r_{in}. Although at first glance the basic concept is almost the same as that of CSS, CCAS can be expected to achieve a better generalization capability than CSS. The reason is that the subsampled pixel values in CCAS correspond to important physical phenomena, in which diffracted light from a mask pattern is propagated concentrically. This feature can also improve the expressive capability of a layout by sampling not only pixel values but by summation of pixels around the sampling area. In Sec. 6 we will further analyze the advantages of this feature, and compare it with the conventional CSS.

5 Optical Proximity Correction Methodologies

In this section, we present a model training approach including a concept of HBM and a Markov Chain Monte Carlo technique. The principal difference between our approach and the conventional linear regression model is that our model is not restricted by linear correlation of model parameters. This indicates that it is expected to realize flexible modeling even of a complicated phenomenon having a large variation in input data or including unknown variables that cannot be measured. In this subsection, we present the concept of the proposed OPC regression, followed by the solution of the parameter estimation technique.

5.1 *Hierarchical Bayes Model*

As mentioned in Sec. 2, the edge displacements vary depending on edge types, such as normal edge, convex edge, concave edge, and line end edge. To simulate the trend of edge displacement, our proposed model considers the regression-based OPC problem with a generalized linear mixed model (GLMM) regarding the edge types as random effects that can express an effect of variance of supervised data. In addition, the GLMM is modeled by a Bayesian approach to deal with many random effects. Figure 5 shows the basic concept of our proposed model where data, edge displacement, follows the Normal distribution of center y and variance σ_y. The

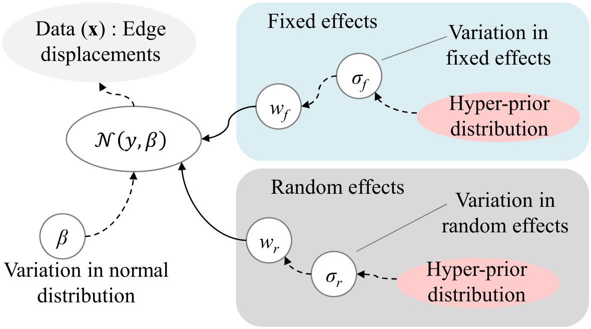


Fig. 5 Concept of the proposed HBM.

model parameters w_{fj}, $\forall\ j \in D$ correspond to fixed effects that are the common effects for all edges. w_{rj}, $\forall\ j \in D$ indicate random effects that are identification effects assigned to edge types. σ_f and σ_r are the variances of fixed effects and random effects, respectively. Although the Bayes inference technique requires some prior information, it is not provided in many practical cases. We therefore propose a regression model using noninformative prior distribution for unknown variables. Finally, our proposed model can be written as follows:

$$p(\mathbf{y}|\mathbf{x},\theta) = \mathcal{N}(y(\mathbf{x},\theta),\sigma_y), \tag{4}$$

$$y(\mathbf{x},\theta) = w_{f_0} + w_{r_0} + \sum_{j=1}^{D}(w_{f_j} + w_{r_j})x_j, \tag{5}$$

where $p(\mathbf{y}|\mathbf{x},\theta)$ is the probability of edge displacements $\mathbf{y}$ given feature vectors $\mathbf{x}$ and model parameters θ, N is the normal distribution of center y and variance σ_y, and D is the total number of dimensions. The prior distributions are defined as follows:

$$w_{f_i} \sim \mathcal{N}(0,\sigma_f), \tag{6}$$

$$w_{r_i} \sim \mathcal{N}(0,\sigma_r), \tag{7}$$

$$\sigma_y \sim \mathcal{U}(0,10^4), \tag{8}$$

where $\mathcal{N}(0,\sigma)$ is the normal distribution of center 0 and variance σ, and $\mathcal{U}(0,10^4)$ is the uniform distribution in the interval $(0,10^4)$. Because our model assumes there is no prior knowledge about the variances of prior distributions, the model utilizes hierarchical prior distributions as noninformative hyper-priors to consider every possibility of unknown parameters.

$$\sigma_f \sim \mathcal{U}(0,10^4), \tag{9}$$

$$\sigma_r \sim \mathcal{U}(0,10^4). \tag{10}$$

The noninformative hyper-priors that follow the uniform distribution in the interval $(0,10^4)$ indicate that the variance can take any values within the range of 0 and 10^4. Finally, the posterior distribution is derived as follows:

$$p(\theta|\mathbf{y}) = \prod_{i=1}^{N}\prod_{j=0}^{D} p(y_i|\theta_j)p(w_{f_j}|\sigma_f)p(w_{r_j}|\sigma_r)p(\sigma_y)p(\sigma_f)p(\sigma_r), \tag{11}$$

where N is the total number of samples, and θ include all model parameters, such as fixed effects, random effects, variance of y, and variances of fixed effects and random effects. Because the posterior distribution includes hierarchical prior distributions, this model is called the HBM. It is difficult to solve all parameters using the likelihood estimation method owing to the complexity of integral computation. Therefore, we estimate all parameters using a sampling technique.

5.2 *Markov Chain Monte Carlo*

MCMC is a method of estimating model variables by sampling parameters from a posterior distribution, or a parameter distribution proportional to the likelihood. Generally, it is difficult to apply a likelihood estimation method, which is a method of estimating model variables that maximizes the likelihood function, to a complicated model such as Eq. (11) because the method must compute the likelihood in the combination of all model parameters. In contrast, it can be expected that appropriate parameters can be obtained by MCMC since the estimation performance is less susceptible to the model complexity.[12]

We briefly describe the flow of MCMC as follows. In this algorithm, a target distribution $\pi(\theta)$, a distribution of model parameter θ, can be obtained from a probability distribution $q(\theta')$ called the proposal distribution. The model parameters are estimated by the following steps: suppose the target distribution $\pi(\theta|\mathbf{x})$ of parameter θ given data $\mathbf{x}$, where $\theta = (\theta_1,\dots,\theta_m)$ and m is the total number of parameters. We first give the initial values of θ as $\theta^{(0)} = (\theta_1^{(0)},\dots,\theta_m^{(0)})$, where $\theta^{(t)}$ indicates a random variable at a time point t. Then, we repeat the following steps for $t = 0,1,\dots,k$:

1. Give the initial variables $\theta^{(0)} = (\theta_1^{(0)},\dots,\theta_m^{(0)})$.
2. Repeat sampling for $t = 0,1,\dots,k$.
   a. Generate θ' from $q(\theta',\theta^{(0)}|\mathbf{x})$.
   b. Generate u from $\mathcal{U}(0,1)$ and select $\theta^{(t+1)}$ based on the following equation:

$$\theta^{(t+1)} = \begin{cases} \theta', & \text{if } u \le \alpha(\theta^{(t)},\theta|\mathbf{x}), \\ \theta^{(t)}, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathcal{U}(0,1)$ is the uniform distribution in the interval (0, 1), α is the selection rate defined by

$$\alpha(\theta^{(t)},\theta|\mathbf{x}) = \min\left\{1, \frac{\pi(\theta'|\mathbf{x})q(\theta',\theta^{(t)}|\mathbf{x})}{\pi(\theta^{(t)}|\mathbf{x})q(\theta^{(t)},\theta'|\mathbf{x})}\right\}. \tag{13}$$

This is known as the Metropolis–Hastings algorithm.[12] Since MCMC requires many iterated calculation to sample stable parameters, evaluating convergence of MCMC is important for measuring the performance of estimated samples. To monitor the convergence of MCMC, this paper uses

$\hat{R}$ index, which is proposed by Gelman and Rubin.[13] This can be computed based on the variance in the Markov chains and defined as follows:

$$\hat{R} = \sqrt{\frac{\widehat{\mathrm{var}}(\psi|y)}{W}}, \tag{14}$$

$$\widehat{\mathrm{var}}(\psi|y) = \frac{k-1}{k}W + \frac{1}{k}B, \tag{15}$$

where k is the number of iterations, B is the between-chain variance and W is the within-chain variance defined by

$$B = \frac{k}{m-1}\sum_{j=1}^{m}(\bar{\psi}_j - \bar{\psi})^2, \tag{16}$$

$$W = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{k-1}\sum_{i=1}^{k}(\bar{\psi}_{ij} - \bar{\psi}_j)^2\right], \tag{17}$$

where m is the number of chains and ψ is the sampled parameter. It is shown that $\hat{R}$ can evaluate the convergence of the sampling results in actual problems.[14] Empirically, if every $\hat{R}$ is smaller than 1.1, it can be concluded that the sampling results are fully converged.[14]

MCMC can estimate reasonable model parameters even in high-dimensional parameter space by constructing a Markov chain. Furthermore, model parameters are estimated without a local solution because of a stochastic process even if the likelihood functions include multi-peak distributions. Therefore, it can be expected that the parameter distributions nearly equaling to the true distributions are obtained by iterative sampling.

6 Experimental Results

The proposed methodologies are implemented in C++ and Python on a Linux machine with eight 3.4 GHz CPUs and 32 GB memory. Calibre[15] is used to perform lithography simulation with wavelength $\lambda = 193$ nm and $NA = 1.35$. Two 32 nm node industrial chips in metal routing layer, layout A and layout B, are applied in the experiment. The areas of the layouts A and B are 9291.37 $\mu\mathrm{m}^2$ and 11702.20 $\mu\mathrm{m}^2$, respectively. In the following, we conduct four experiments related to feature selection, model training, comparison with linear/nonlinear regression, and comparison of model-based OPC.

6.1 *Feature Selection*

In the first experiment, we evaluate the effectiveness of the proposed layout feature extraction. We compare the RMSE values of the proposed CCAS feature and the CSS feature[9] on a testing layout with different feature parameters. Here, linear regression algorithm is selected as the regression model, which is trained with a part of supervised data including all edge types and consists of 5000 random samples from the training layout A. The RMSEs (nm) on the testing layout are computed using 5000 samples extracted from the layout B. Figure 6 compares RMSE values, where parameter r_{in} is ranging from 50 to 500 nm, and parameter l is set to 1.0 μm. Figure 7 compares RMSE values, where parameter l is ranging from 0.5 to 1.4 (μm), and parameter r_{in} is set to 150 nm.

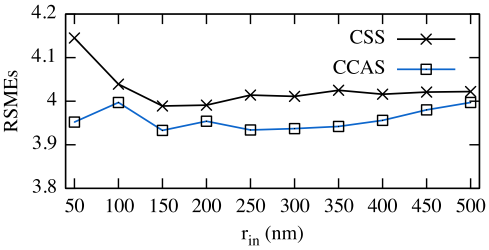


Fig. 6 Comparison between CCAS and CSS on different parameter r_{in} ($l = 1.0$ μm).

From Figs. 6 and 7, we can see that the model trained by the proposed CCAS can always achieve less RMSE values. That is, the CCAS feature shows a better predictive performance than the CSS feature. Therefore, in the following experiments we set $l = 1.0$ μm and $\mathrm{r_{in}} = 150$ nm, which can provide the best prediction accuracy.

6.2 *Sampling Results*

In the second experiment, we train the proposed HBM for OPC regression Eq. (11). As mentioned in Sec. 5, HBM includes many parameters. Here the layout pattern is sampled 2.5 nm/1 pixel resulting in 400 × 400 pixel binary map to fit the grid size of the optical model. Since the total number of dimensions is 257, there are 1293 parameters in total, e.g., 258 fixed effects including bias parameter w_{f_0}, 1032 (258 × 4) random effects including bias parameters w_{r_0}, two hierarchical hyper-priors σ_f, σ_r, and one variance in normal distribution σ_y. All parameters are estimated using the MCMC algorithm described in Sec. 5. For the MCMC parameters, we set the number of iterations to 5000, the number of chains to four, the burn-in number to half of the iterations, and the amount of thinning to 10. Since it is difficult to introduce all estimation results, we only show the results of hierarchical parameters peculiar to HBM, σ_f and σ_r, as shown in Fig. 8. In this figure, some of the sampling results are shown on the left, and the estimated probability density functions of sampled parameters are indicated on the right.

Although the sampling results of random effects are similar, they are far from being the same. The result shows that there is a clear difference among edge types and these hidden relationships can be estimated by HBM without any prior knowledge. Furthermore, the figure indicates that the parameters are well converged because all $\hat{R}$ are less than 1.1.

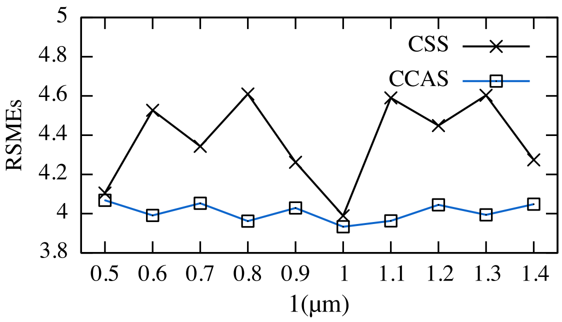


Fig. 7 Comparison between CCAS and CSS on different parameter l ($r_{\mathrm{in}} = 150$ nm).

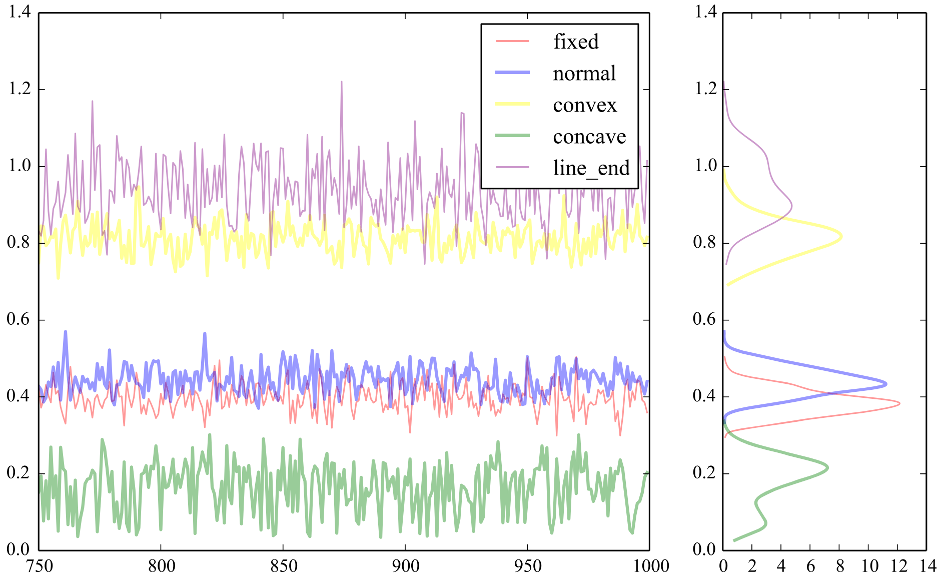


Fig. 8 Sampling results of hidden variables. $\hat{R}$ values of σ_f, σ_r(normal), σ_r(convex), σ_r(concave), and σ_r(line-end) are 1.09, 1.01, 0.99, 0.98, and 1.00, respectively.

6.3 *Comparison with Linear Regression and Nonlinear Regression*

In the third experiment, we compare the proposed HBM with linear regression method and nonlinear regression method, respectively. A SVR algorithm is implemented as the nonlinear regression method. For both linear and nonlinear regression methods, we consider two sample sets, unified sample set (labeled as "**all**") and separated sample set (labeled as "**sep**"). The reason is that separate sample set may have better predictive performance than a unified sample set.[9] The proposed HBM method utilizes a unified sample set. For each regression method, Tables 1 and 2 list the predictive RMSE values on training layout and testing layout, respectively.

From Table 1, we can see that the SVR model, which is a nonlinear regression, has relatively better RMSE for training data set compared with other algorithms. However, from Table 2 we can see that it is difficult to achieve high prediction accuracy for unknown data with the nonlinear model, which indicates a typical over-fitting issue. Our proposed HBM achieves the best prediction accuracy in testing layout.

We further compare the difference of prediction accuracy between HBM and the linear regression model when the training sample number is changed. As shown in Fig. 9, in the linear regression model, the prediction accuracy drastically deteriorates if the number of training data decreases. In contrast, HBM can prevent degradation of prediction accuracy even with small numbers of training data. The reason is that HBM learns hidden variables corresponding to variance components included in given data. From the above results, we can get the conclusion that, through using HBM it is possible to achieve a robust OPC regression model with good predictive performance even when the number of samples is relatively small.

In the comparison experiment in Fig. 9, calibration times on single core for all, middle, and small models are 46 (min), 106 (h), and 613 (h), respectively. It should be noted that although HBM calibrations take a lot of time since the method uses a scheme of MCMC, prediction time of HBM is almost the same as the linear regression-based model.

Table 1 RMSE comparison on training layout.

Edge type	LR (all)	LR (sep)	SVR (all)	SVR (sep)	HBM
All	3.575	3.154	0.518	0.506	3.179
Normal	2.852	2.696	0.534	0.534	2.703
Convex	5.109	4.460	0.505	0.429	4.466
Concave	5.166	3.667	0.150	0.150	3.860
Line end	4.666	3.428	0.511	0.509	3.581
Average	4.274	3.481	0.444	0.426	3.558

Table 2 RMSE comparison on testing layout.

Edge type	LR (all)	LR (sep)	SVR (all)	SVR (sep)	HBM
All	3.701	3.664	8.875	7.424	3.492
Normal	2.888	2.778	6.765	6.361	2.747
Convex	5.566	5.166	10.419	9.744	5.114
Concave	5.172	4.670	6.506	6.497	4.380
Line end	4.697	6.262	20.088	11.252	4.980
Average	4.405	4.508	10.530	8.256	4.143

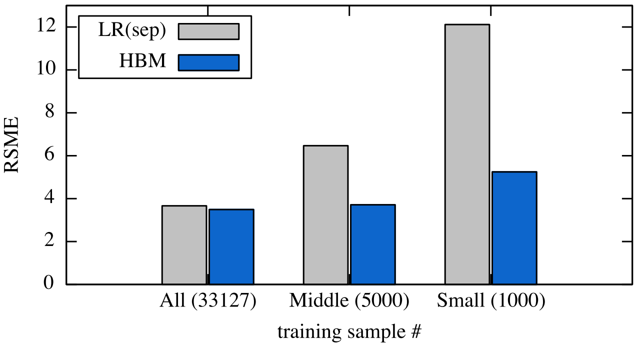


Fig. 9 Performance comparison on different training sample number.

6.4 *Comparison with Model-based Optical Proximity Correction*

In this last experiment, we compare the proposed HBM model with conventional model-based OPC. We measure correction accuracy based on an edge displacement error (EPE) distribution. Figure 10 shows the EPE distribution comparison between the model-based OPC and the proposed HBM. In the figure, the blue line, the green line with plus marker and the red line with down triangle marker show the EPEs at 2nd, 6th, and 10th iteration in model-based OPC, respectively. The aqua line with x marker illustrates the EPE distribution in the predicted results by HBM without any model-based iteration.

Although HBM (aqua) result looks even better in terms of std. and variation in Fig. 10, actually, MBi10 (red) has better std. and variation than HBM (aqua). To make the conclusion more clear, a subgraph is added to zoom into the scope with negative EPE values. We can see that some predictions of HBM have reverse direction, thus conventional model-based OPC is still a must to achieve good OPC results. The major reason of such reverse directionality is twofold. (1) Layout feature: the proposed CCAS method has better performance than conventional method. However, there is a possibility that the CCAS is not necessarily appropriate for representing layout characteristics. (2) Design of HBM: there may be several hidden hyper-parameters which can control other unknown variations in hierarchical Bayes modeling. These are subjects for future works.

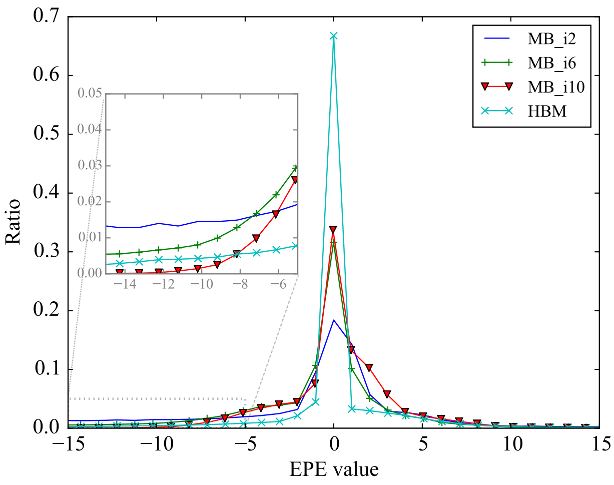


Fig. 10 EPE distributions.

Table 3 Comparison with model-based OPC.

	MB(i2)	MB(i4)	MB(i6)	MB(i8)	MB(i10)	HBM
Variance	71.094	52.208	35.086	13.131	9.768	14.514
Std. dev.	8.432	7.226	5.923	3.624	3.125	3.810
Mean	−3.036	−2.214	−1.521	−0.507	0.216	−0.196
Median	−0.292	−0.329	−0.250	−0.158	0.039	0.000

Table 3 lists the comparison of variance, std. dev., mean, and median between model-based OPC and our regression model. From the table, we can see that the regression results by HBM can achieve almost the same variance as the result of 8th iteration in model-based OPC. This indicates that the number of iterations can be reduced to two by using the regression results as an initial input of model-based OPC. Furthermore, since the regression model achieves the best mean and median EPEs, further improvement of prediction accuracy can be expected by adding several hidden variables to HBM, tuning MCMC parameters or adjusting model-based OPC recipe.

Figure 11 shows OPC results on a one-dimensional layout. The gray features are the target pattern. The red line and the blue line indicate the model-based OPC result (after 10 iterations) and our proposed HBM-based OPC result, respectively. From the figure, we can see that the OPC outputs are very similar for both methods.

Figure 12 further shows several OPC results on two-dimensional layouts. We can see that even for complex two-dimensional layouts, the proposed HBM-based OPC can achieve very comparable solutions against conventional model-based OPC. In the figure, overlapping contours are not shown to easily recognize a difference of edge movements between model-based OPC and HBM. It should be noted that several predictions have reverse direction in Figs. 12(c) and 12(d). This result and the discussions on EPE distributions (see Fig. 10) show that there is still room for improvement of prediction accuracy by designing more hidden hyper-parameter which controls unknown effects from adjacent edges.

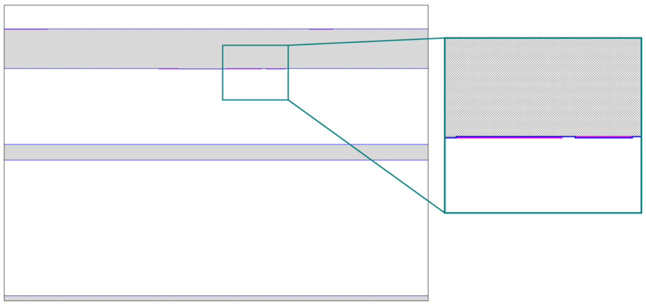


Fig. 11 OPC results on a one-dimensional layout: the red line is model-based OPC, while the blue line is HBM-based OPC.

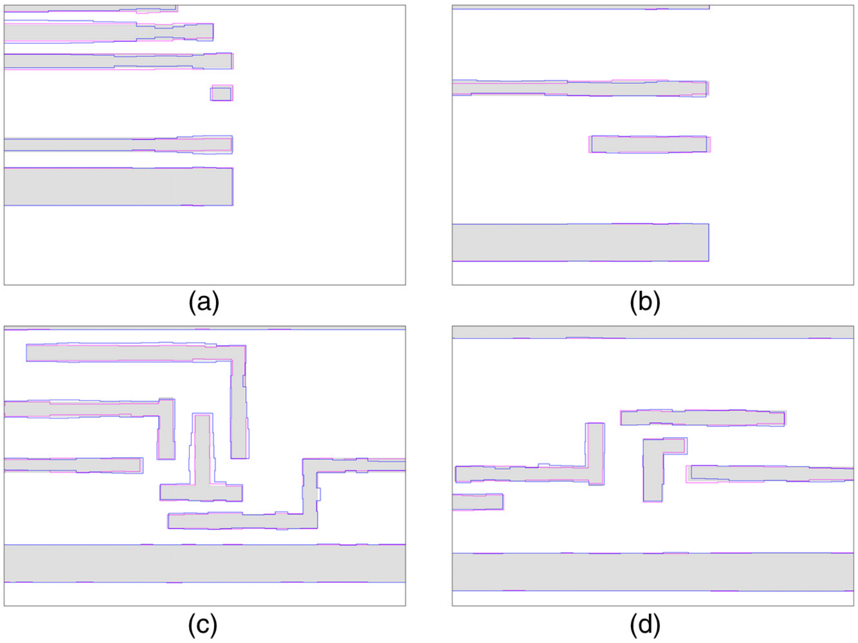


Fig. 12 OPC results on two-dimensional layouts: the red line is model-based OPC, while the blue line is HBM-based OPC. (a) pattern 1, (b) pattern 2, (c) pattern 3, and (d) pattern 4.

7 Conclusions

This paper proposes a new regression-based OPC technique based on a HBM. By applying HBM, flexible modeling is realized without being restricted by linearity of model parameters or the number of supervised samples, which can improve predictive performance of a regression model with a small number of data. Markov Chain Monte Carlo is able to estimate all model parameters including unknown variables that correspond to variance components included in given data. The experimental results show that the proposed HBM can achieve very comparable results with MB(i8), which means the first eight iterations can be replaced by the proposed HBM. Then the iteration number of conventional MBOPC can be reduced from ten to two. In other words, our method is promising to dramatically reduce the cost of process development.

Acknowledgment

This work was supported in part by National Science Foundation (NSF), Semiconductor Research Corporation (SRC), and Chinese University of Hong Kong (CUHK) Direct Grant for Research.

References

1. V. Bakshi, *EUV Lithography*, Vol. **178**, SPIE Press, Bellingham, Washington (2009).
2. H. C. Pfeiffer, "New prospects for electron beams as tools for semiconductor lithography," *Proc. SPIE* **7378**, 737802 (2009).
3. T. Higashiki, T. Nakasugi, and I. Yoneda, "Nanoimprint lithography for semiconductor devices and future patterning innovation," *Proc. SPIE* **7970**, 797003 (2011).
4. Y. Seino et al., "Contact hole shrink process using directed self-assembly," *Proc. SPIE* **8323**, 83230Y (2012).
5. D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **32**(10), 1453–1472 (2013).
6. N. B. Cobb and Y. Granik, "OPC methods to improve image slope and process window," *Proc. SPIE* **5042**, 116–125 (2003).
7. S. Miyama, K. Yamamoto, and K. Koyama, "Large-area optical proximity correction with a combination of rule-based and simulation-based methods," *Jpn. J. Appl. Phys.* **35**(12S), 6370 (1996).
8. N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *J. Opt.* **12**(4), 045601:1–045601:9 (2010).
9. A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Trans. Semicond. Manuf.* **21**(2), 263–271 (2008).
10. R. Luo, "Optical proximity correction using a multilayer perceptron neural network," *J. Opt.* **15**(7), 075708 (2013).
11. K.-S. Luo et al., "SVM based layout retargeting for fast and regularized inverse lithography," *J. Zhejiang Univ.—Sci. C* **15**(5), 390–400 (2014).
12. W. R. Gilks, *Markov Chain Monte Carlo*, John Wiley & Sons Ltd., New Jersey (2005).
13. A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statist. Sci.* **7**(4), 457–472 (1992).
14. S. P. Brooks and A. Gelman, "General methods for monitoring convergence of iterative simulations," *J. Comput. Graph. Stat.* **7**(4), 434–455 (1998).
15. Mentor Graphics, "Calibre verification user's manual," (2008).

Tetsuaki Matsunawa received his PhD in computer science from the University of Tsukuba in 2008. He joined Toshiba Corporation in 2008 and has been working in the area of optical lithography. He visited the University of Texas at Austin as a visiting scholar from 2013 to 2015. His current research interests include design for manufacturability and machine learning algorithms with applications in computational lithography.

Bei Yu is an assistant professor at the Department of Computer Science and Engineering, the Chinese University of Hong Kong. His research interests include design for manufacturability and optimization algorithms with applications in CAD. He received EDAA outstanding dissertation award in 2014, SPIE education scholarship in 2013, IBM PhD scholarship in 2012, best paper awards at ICCAD'13 and ASPDAC'12, and three other best paper award nominations.

David Z. Pan is a professor at the Department of Electrical and Computer Engineering, University of Texas at Austin. His research interests include nanometer IC design for manufacturability/reliability, new frontiers of physical design, and CAD for emerging technologies. He has published over 200 technical papers, and is the holder of eight U.S. patents. He has received many awards, including SRC technical excellence award and 11 best paper awards, among others. He is an IEEE fellow.