

Optical Proximity Correction with Linear Regression

Allan Gu and Avidesh Zakhori, Fellow, IEEE

Department of Electrical Engineering and Computer Sciences

University of California at Berkeley, CA 94720, USA

Email: {agu, avz}@eecs.berkeley.edu

Abstract

An important step in today's Integrated Circuit (IC) manufacturing is optical proximity correction (OPC). In model based OPC, masks are systematically modified to compensate for the non-ideal optical and process effects of optical lithography system. The polygons in the layout are fragmented, and simulations are performed to determine the image intensity pattern on the wafer. If the simulated pattern on the wafer does not match the desired one, the mask is perturbed by moving the fragments. This iterative process continues until the pattern on the wafer matches the desired one. Although OPC increases the fidelity of pattern transfer to the wafer, it is quite CPU intensive due to the simulations performed at each iteration. In this paper, linear regression techniques from statistical learning are used to predict the fragment movements. The goal is to reduce the number of iterations required in model based OPC by using a fast, computationally efficient linear regression solution as the initial guess to model based OPC. Experimental results show that fragment movement predictions via linear regression model significantly decrease the number of iterations required in model based OPC, thereby decreasing the product development time in I.C. design and manufacturing.

I. INTRODUCTION

An important step in today's IC manufacturing is optical proximity correction (OPC); it is nearly impossible to fabricate modern IC designs without OPC. OPC modifies the mask to compensate for the non-ideal optical and process effects of optical lithography system. As seen in Figure 1(a), the layout pattern without OPC does not transfer properly onto the wafer, i.e. there is a line end shortening and

Correspondence: avz@eecs.berkeley.edu, 510-643-6777

rounding on the wafer. However, the same layout pattern with OPC transfers more accurately onto the wafer as shown in Figure 1(b). Although OPC increases the fidelity of pattern transfer to the wafer, it is quite CPU intensive.

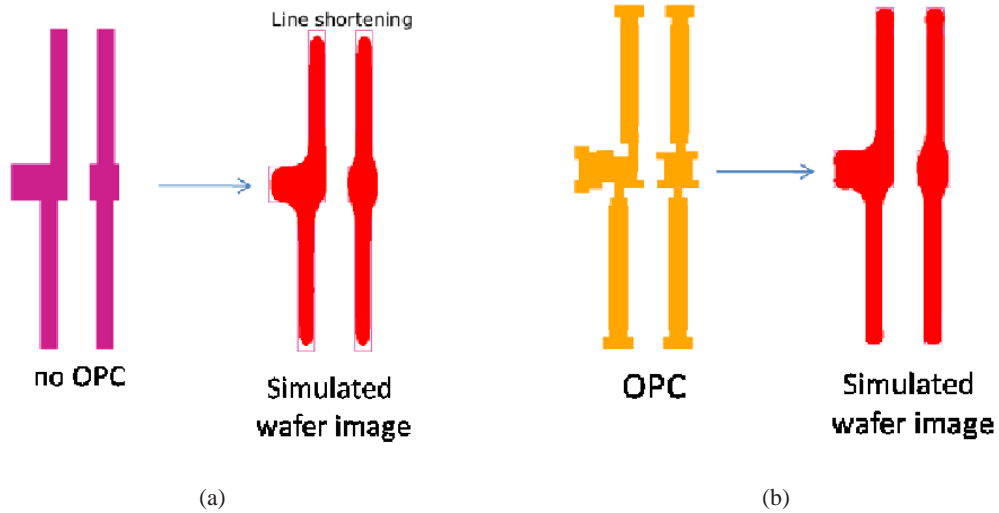


Fig. 1. (a) Layout pattern without OPC and the simulated image on the wafer; (b) layout pattern with OPC and the simulated image on the wafer.

One approach to OPC is to decompose the mask into small pixels and to optimize each pixel [1]–[4]. Since this results in complex masks that are difficult to manufacture, current OPC algorithms decompose the mask into edges and corners and optimize their locations. There are two types of edge based OPC: rule based and model based. In rule based OPC, the lithography engineer creates different experiments to determine the corrections that are needed to compensate for the non-ideal effects of the optical lithography system and the resist. Rules are empirically generated based on the geometrical properties of the layout patterns [5], [6]. Unlike rule based OPC, model based OPC [7]–[9] typically uses simulations in an iterative manner. The layout consisting of polygon edges is fragmented, and at each iteration corresponding to a particular position of the fragments, simulations are performed to determine the image intensity on the wafer. If the simulated image on the wafer does not match the desired one, then the mask is further perturbed by moving the fragments. This process continues until the simulated image on the wafer matches the desired one. Although model based OPC is more robust than rule based OPC, it is also much more computation intensive due to the simulations that are performed at each iteration. For instance, performing full chip model based OPC for modern day VLSI designs can take many days using thousands of CPUs. A good initial estimate of the final mask pattern supplied to the model based

OPC algorithm can significantly reduce the number of iterations required for the algorithm to converge and thereby, reduces the overall OPC run time. In practice, a hybrid solution is usually taken by OPC engineers with simple rule based OPC being followed by model based OPC [10]–[13].

In this paper, we propose a method to modify a layout using linear regression so as to decrease the number of iterations required for model based OPC to converge, and to decrease the OPC run time and overall product development time. There has been prior works on using neural networks to correct for non-idealities in lithography systems. Frye *et al.* [14] use neural networks to compensate for electron scattering effects in E-beam lithography systems resulting in significant reduction in computation time as compared to iterative algorithms. Jedrasik [15] has proposed a neural network approach for one step OPC. Recently, Huang *et al.* [16] have proposed a similar idea as the one presented here to increase model based OPC convergence rate. Similar to the prior work, Huang *et al.* train a neural network to map the fragment movements. However, they have only tested their method on a single polygon, and do not consider more complex patterns with many polygons.

The outline of the paper is as follows: Linear regression is presented in Section 2. Section 3 describes the training and evaluation methodology used to select the best linear regression model. Section 4 presents the prediction results on different portions of two 90nm layouts using the linear regression model selected in Section 3. In Section 4, we show that the fragment movements obtained via linear regression can be used as initial conditions for model based OPC to reduce the number of iterations. Finally, conclusions and future work are presented in Section 5.

II. LINEAR REGRESSION

Regression is a statistical technique [17] which models the dependence of the output y on the input features \vec{x} . Linear regression model assumes the output, y , is linearly dependent on the input feature \vec{x} plus some noise. This can be written as $y = \vec{\beta}^T \vec{x} + \epsilon$, where ϵ is assumed to be a zero mean additive Gaussian noise with variance σ^2 . $\vec{\beta}$ is the parameter vector, which specifies how much each component of \vec{x} contributes to the output, y . This means that $P(y|\vec{\beta}, \sigma^2, \vec{x}) = N(\vec{\beta}^T \vec{x}, \sigma^2)$. Therefore, given \vec{x} , the best estimate of y is $E[y|\vec{x}] = \vec{\beta}^T \vec{x}$.

In order to estimate y for a given \vec{x} , $\vec{\beta}$ is needed. However, $\vec{\beta}$ is usually unknown, and needs to be estimated through a training process. Given N observation pairs, $\{(y_i, \vec{x}_i), i = 1, 2, \dots, N\}$, it is possible to estimate the value of $\vec{\beta}$ as the one that minimizes the $\sum (y_i - \vec{\beta}^T \vec{x}_i)^2$, or equivalently

$$\vec{\beta} = \arg \min_{\vec{\beta}} \left\| \mathbf{X} \vec{\beta} - \vec{y} \right\|^2 \quad (1)$$

where each input feature vector \vec{x}_i is a row in the matrix \mathbf{X} and each output, y_i , is a component in the vector \vec{y} . It can be easily shown [18] that $\vec{\beta} = (\mathbf{X}^T \mathbf{X})^{-1}(\mathbf{X}^T \vec{y})$ is the solution to the least square problem in Equation 1.

III. TRAINING AND EVALUATION

In this section, we determine the best input feature and its associated dimension. Mentor Graphics CalibreTM is used to perform model based OPC using a vector optical model with wavelength $\lambda = 193\text{nm}$ and $\text{NA} = 0.85$ for two 90nm IC designs. An annular aperture with $\sigma = 0.88/0.44$ and a VT5 resist model is used. The surrounding $2\mu\text{m}$ by $2\mu\text{m}$ layout pattern for each fragment is captured and used to derive its movement using our proposed method. We choose the $2\mu\text{m}$ by $2\mu\text{m}$ layout pattern surrounding each fragment since the spatial influence of the optical model has a diameter of $1.28\mu\text{m}$, and the interaction diameter in the resist model is $1.8\mu\text{m}$. The layout pattern is sampled at 5 nm per pixel resulting in a 400 pixel by 400 pixel binary bitmap. The 5nm sampling is chosen because the optical model has a 5 nm optical grid size. Figure 2 shows a fragment that is perturbed by OPC software in black and its surrounding $2\mu\text{m}$ by $2\mu\text{m}$ layout pattern. In addition, fragments are separated into normal edge, convex corner, and concave corner fragments as shown in Figure 3. Convex corner fragments are those that form a convex corner with other fragments; concave corner fragments are those that form a concave corner with other fragments; all other fragments are normal edge fragments corresponding to simple edges. For example, the two fragments shown in green in Figure 3 are convex corner fragments, and the two fragments in blue in Figure 3 are concave corner fragments.

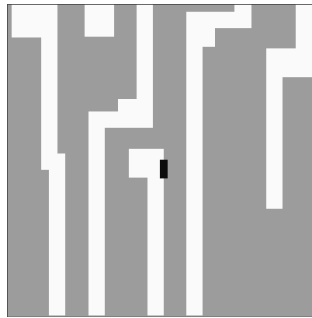


Fig. 2. Example of $2\mu\text{m}$ by $2\mu\text{m}$ layout pattern. The fragment of interest is in black.

In optical lithography, the maximum spatial frequency is $\frac{NA}{\lambda}$ where NA is the numerical aperture of the lens and λ is the wavelength of the illumination source. It has been shown that different resolution enhancement techniques [19] can at most increase the maximum spatial frequency to $\frac{2NA}{\lambda}$ [20]. As such,

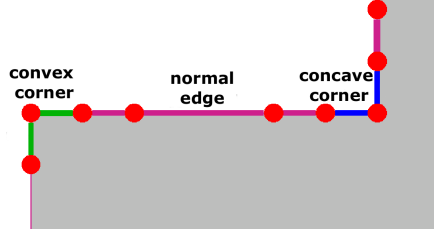


Fig. 3. Examples of normal edge, convex corner, and concave corner fragments. Normal edge fragments are in violet, convex corner fragments are in green, and concave corner fragments are in blue.

patterns with sharp corners are not physically feasible, and therefore, we choose to use low pass filter layout patterns as features in our linear regression model to predict the fragment movements. Specifically, we filter the $2\mu\text{m}$ by $2\mu\text{m}$ layout pattern with a Gaussian low pass filter with cut off frequency of $\frac{2NA}{\lambda}$. Figure 4 shows the original layout pattern and the resulting filtered pattern that we use for training and evaluation.

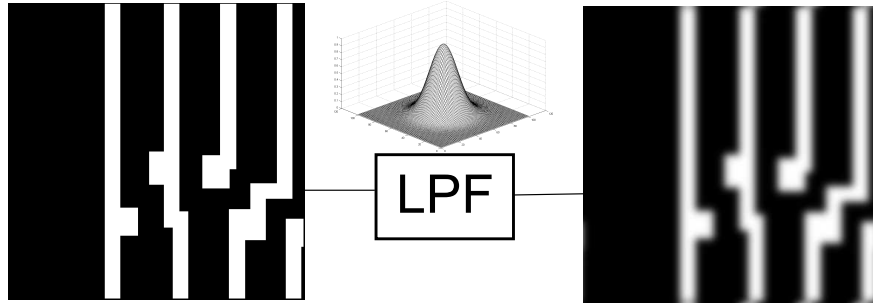


Fig. 4. Original layout pattern and the resulting filtered pattern.

The optimal fragment movement depends on the $2\mu\text{m}$ by $2\mu\text{m}$ pattern. However, it is impractical to use all the layout pixels as the input feature vector; specifically, the 400 pixel by 400 pixel pattern translates into a feature vector in \mathbb{R}^{160000} . For this high dimensional vector, it is very difficult to collect enough sample patterns and the corresponding fragment movements in order to ensure that the resulting matrix $X^T X$ is invertible. In addition, evaluating the value of $\vec{\beta}$ in the linear regression model is quite computation intensive. However, because of the low pass operation, the pattern mainly consists of low frequency components, and as such, only few frequency domain coefficients, such as Discrete Cosine Transform (DCT), are required to approximate it. Furthermore, since optical and process proximity effects decrease with distance, we can also sub-sample the 400 pixel by 400 pixel patterns more densely near

the fragment of interest.

In the remainder of this section, experiments are performed to determine the optimal input feature such as space domain pixels or frequency domain DCT coefficients, as well as the number of components for each type of feature. Additionally, we determine whether multiple regression models are needed for multiple types of fragments such as normal edge, and convex and concave corners. In Section IV, we present prediction results on new data sets using the optimum input features determined in this section.

We use the root mean square prediction error (RMSPE) as a metric to compare the performance of the various approaches. The RMSPE is defined as

$$RMSPE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y})^2} \quad (2)$$

where y_i is the fragment movement determined by model based OPC, and \hat{y} is the predicted fragment movement obtained in our proposed linear regression model. The smaller the RMSPE value, the more effective is the linear regression model at predicting the fragment movement. If the RMSPE is 0, then every predicted fragment movement is exactly equal to the fragment movement obtained via model based OPC. In this paper, the RMSPE is calculated on new test data sets that are not used during the training process to estimate the parameter vector $\vec{\beta}$.

A. Training

We perform model based OPC on a $100\mu\text{m} \times 100\mu\text{m}$ section of logic layout from design A, and select 4871 pairs of fragment movements and associated patterns among all the available pairs in the $100\mu\text{m} \times 100\mu\text{m}$ section of the layout. These pairs are selected in such a way as to cover all possible values of model based OPC fragment movements in the $100\mu\text{m} \times 100\mu\text{m}$ patch. For example, if model based OPC results in fragment movement of 36nm for any fragments in the $100\mu\text{m} \times 100\mu\text{m}$ patch, we make sure that at least one of the 4871 pairs correspond to model based OPC fragment movement of 36nm. About half of the 4871 pairs, or 2436 pairs are used for training, and the remaining half are used for testing. The 4871 pairs are divided into training and evaluation set in such a way that both sets contain approximately equal number of the same fragment movements. However, if a particular value of fragment movement is associated with only one pattern, then it is placed in the training set to ensure the training set covers the entire range of fragment movements. The 2436 training pairs consist of 625 pairs with normal edge, 847 pairs with concave corner, and 964 pairs with convex corner fragments.

The linear regression model is trained with DCT coefficients from the filtered $2\mu\text{m}$ by $2\mu\text{m}$ layout patterns. The first 200, 300, 400, and 500 DCT coefficients are used as the input features, and the

coefficients are encoded into a feature vector using zig-zag ordering from lowest to highest frequency. As seen in Figure 5, the DC component of the DCT coefficients becomes the first component of the feature vector, and the rest of the feature vector is populated with the low to high frequency coefficients. In addition to the DCT coefficients, each input feature vector also contains 3 binary variables indicating the type of fragment whose movement is being predicted.

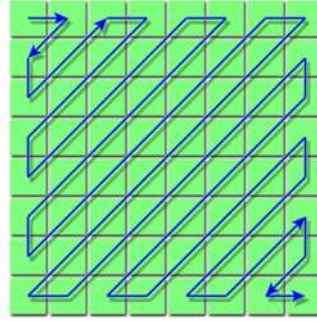


Fig. 5. Zig-Zag ordering of DCT coefficients.

We also train the regression model with sampled pixel values of the filtered layout pattern. A concentric square sampling method as shown in Figure 6 is used to create the feature vectors. Pixel values are sampled at the 4 corners and the mid point of each side of the squares that overlay the pattern. The radius of the concentric squares are $0, 4, 8, 12, \dots, R_{int}, R_{int} + 8, R_{int} + 16, \dots, 200$ pixels respectively where R_{int} controls the sampling density. The regression model is trained with $R_{int} = \{60, 80, 100, 120, 140\}$ resulting in 257, 281, 297, 321, and 337 pixel values respectively. Similar to DCT coefficients, in addition to the pixel values, each input feature vector also contains 3 binary variables indicating the type of fragments whose movement is being predicted.

B. Evaluation

As described earlier, we use 2435 of the 4821 pairs from design A for evaluation. The testing set contains 625 pairs with normal edge, 846 pairs with concave corner, and 964 pairs with convex corner fragments. The linear regression model is trained with a variety of number of DCT coefficients as described earlier. For each pattern, the fragment movement is predicted as $\hat{y} = \vec{\beta}^T \vec{x}$, where \vec{x} is a feature vector containing 3 indicator variables and the DCT coefficients. The RMSPE as defined in Equation 2 is shown in Table I. As seen, the 200 DCT coefficients result in the highest, and 500 DCT coefficients result in the lowest RMSPE at 8.84nm and 5.20nm respectively.

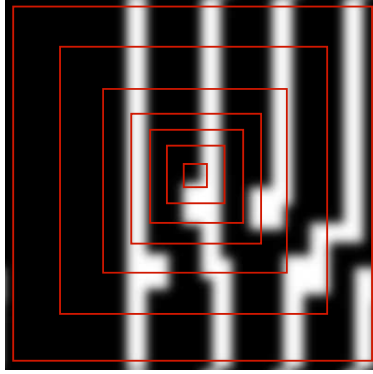


Fig. 6. *Concentric square sampling of the pixel values.*

TABLE I

RMSPE for different number of DCT coefficients. RMSPE is in nm.

# DCT Coeff.	RMSPE
200	8.840
300	7.191
400	6.240
500	5.198

Linear regression model is also trained with input feature vectors containing sub-sampled pixel values on concentric squares. The model is applied on the 2435 test data points, and the RMSPE as defined in Equation 2 is shown in Table II. As seen in the 2nd column of Table II, the largest RMSPE is 3.51 nm when $R_{inner} = 60$ and the smallest RMSPE is 3.44 nm when $R_{inner} = 120$. The largest RMSPE using pixel values as input features is 34% smaller than the smallest RMSPE using DCT coefficients as input features. This means that the pixel values are likely to outperform DCT coefficients as input features. As mentioned earlier, OPC increases the spatial frequency of the resulting patterns on the wafer, and therefore, it is possible that both the low and high frequency DCT coefficients are needed to effectively predict the fragment movements.

Examining the data more closely, we find that different types of fragments possess different movements characteristics as obtained via model based OPC. Naturally, this raises the question as to whether training separate models for each fragment type is likely to improve the performance. Since sub-sampled pixel values have been shown to outperform DCT coefficients, we use them to create separate models for each fragment type. In doing so, we remove the indicator variables indicating the fragment type from the input

TABLE II
RMSPE for different values of R_{inner} . RMSPE is in nm.

R_{inner}	RMSPE
60 (257 values)	3.509
80 (281 values)	3.453
100 (297 values)	3.481
120 (321 values)	3.439
140 (337 values)	3.483

feature vector. The RMSPE for each type of fragment using a combined single regression model on sub-sampled pixel values for different values of R_{inner} are shown in columns 2, 3, and 4 of Table III. Also shown in columns 5, 6, and 7 of Table III are the RMSPE values for normal edge, concave corner, and convex corner fragments using separate regression models. Comparing the columns, it is clear that using separate regression models for each type of fragment results in significantly lower RMSPE than single model for all values of R_{inner} . From Table III, we conclude that the best feature vector corresponds to having a separate model for each fragment type consisting of sub-sampled pixel values with $R_{inner} = 80$.

TABLE III
Comparison of RMSPE for normal edge, concave corner, and convex corner fragments using a combined linear regression model with RMSPE using separate linear regression models. RMSPE is in nm.

R_{inner}	normal (single)	concave (single)	convex (single)	normal (sep.)	concave (sep.)	convex (sep.)	normal % diff.	concave % diff.	convex % diff.
60	3.519	3.895	3.1198	2.893	3.565	2.277	17.8%	8.5%	27.0%
80	3.545	3.721	3.1209	2.434	3.459	2.420	31.3%	7.0%	22.4%
100	3.551	3.799	3.1195	2.587	3.522	2.344	27.1%	7.3%	24.9%
120	3.532	3.676	3.1408	2.763	3.546	2.412	21.8%	3.5%	23.2%
140	3.573	3.737	3.1773	2.932	3.731	2.462	17.9%	0.2%	22.5%

IV. RESULTS

In this section, we use sub-sampled pixel values with $R_{inner} = 80$ as input feature vector to train a different model for normal edge, concave corner, and convex corner fragments using a $60\mu\text{m}$ by $50\mu\text{m}$ portion of design A. Even though the optimum model obtained in Section 3 also corresponds to sub-sampled pixel values with $R_{inner} = 80$, its training set does not necessarily cover the entire range of

fragment movements for every training type. To this end, we collect a subset of all the pairs of layout patterns and their corresponding fragment movements obtained via model based OPC in a $60\mu\text{m}$ by $50\mu\text{m}$ patch of layout A for training purpose. Specifically, we collect 2675 pairs of pattern and movement with normal edge, 2784 pairs with concave corner, and 2680 pairs with convex corner fragments in such a way to ensure the training data covers the range of fragment movements for each type of fragment. The resulting parameter vector, $\vec{\beta}$, is used to test on another section of logic layout from design A, as well as a section of logic layout from design B.

A. Comparison with model based OPC

The model is tested on a different $70\mu\text{m}$ by $50\mu\text{m}$ section of logic layout from design A, which has 27432 normal edge, 2796 concave corner, and 8203 convex corner fragments. As shown in Figure 7, the predictions via linear regression model shown in yellow follows the fragment movements obtained via model based OPC shown in black reasonably well. Table IV shows that the smallest RMSPE is 2.125 nm for convex corner fragments, and the largest RMSPE is 2.876 nm for concave corner fragments. Furthermore, as seen in the 4th column of Table IV, for all fragment types, more than 93% percent of the predicted fragment movements are in the same direction as the model based OPC fragment movement. Figure 8 shows the cumulative distributive function (CDF) of the absolute prediction errors for the three fragment types. As seen, more than 63%, 70%, and 77% of the absolute prediction errors are less than 2 nm for concave corner, normal edge, and convex corner fragments respectively.

TABLE IV

RMSPE and percentage of predicted fragment movement in the same direction as model based OPC fragment movement for the three types of fragments. RMSPE is in nm. The data are from a section of design A.

	# of fragments	RMSPE	% Right Dir.
normal	27432	2.494	93.7
concave	2796	2.876	95.3
convex	2784	2.125	97.9

B. Comparison with model based OPC on a different layout

The linear regression model is also tested on a $18\mu\text{m} \times 16\mu\text{m}$ section of logic layout from a different IC design B, with 4055 normal edge, 455 concave corner, and 995 convex corner fragments. Figure 9 shows the predicted movement obtained via linear regression shown in yellow and the corresponding movement

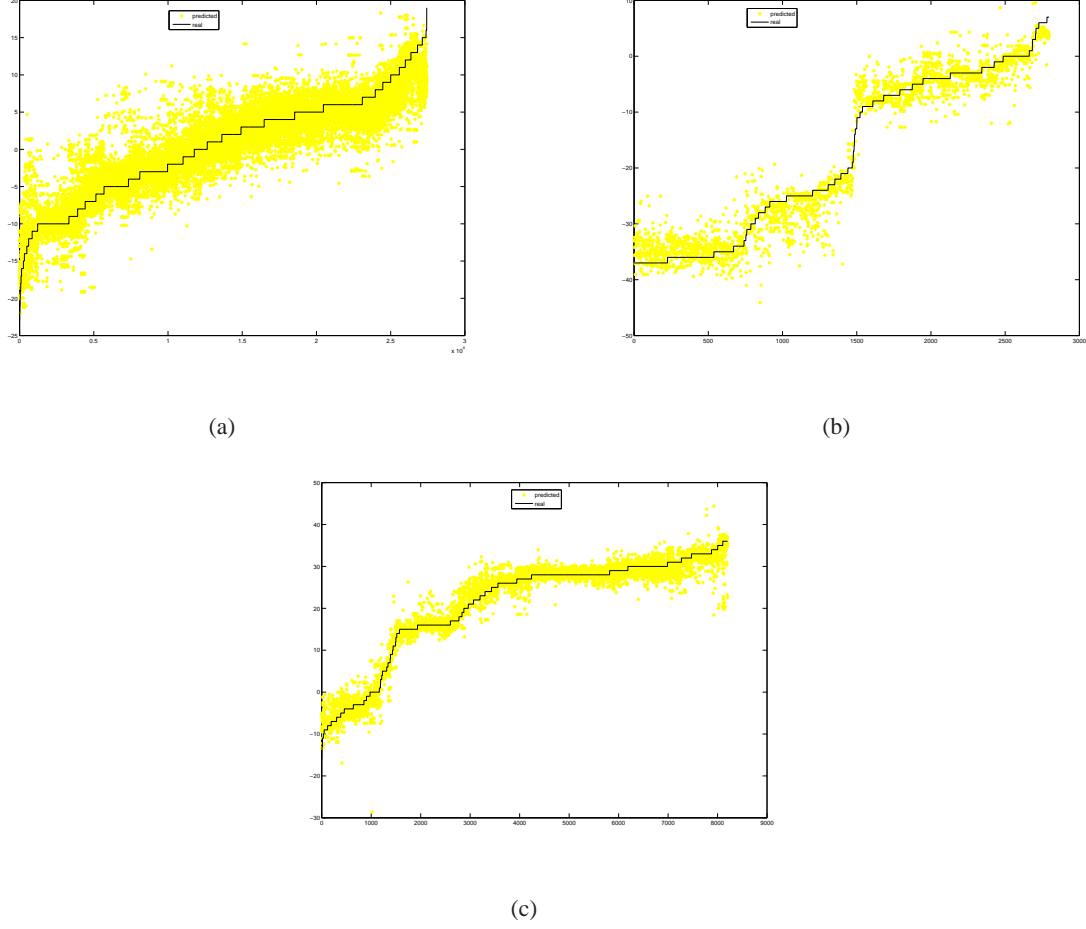


Fig. 7. The model based OPC fragment movement and the predicted fragment movement using separate linear regression models for each type of fragment on a section of layout from design A. (a) normal edge fragment; (b) concave corner fragment; (c) convex corner fragment. The yellow indicates predicted movement from linear regression, and the black shows the movement obtained from model based OPC software.

determined by model based OPC shown in black. It can be seen from Figure 9 and Table V that the predicted fragment movements match the edge movements obtained via model based OPC reasonably well. In particular, as shown in column 4 of Table V, for all fragment types, over 90% of the predicted fragment movements are in the same direction as the fragment movements generated by model based OPC. Furthermore, as shown in column 3 of Table V, the largest RMSPE is 3.94 nm for concave corner fragments. Figure 10 shows that more than 63% of the absolute prediction errors are less than 2 nm for convex corner and normal edge fragments, and more than 42% of the absolute prediction errors are less than 2 nm for concave corner fragments.

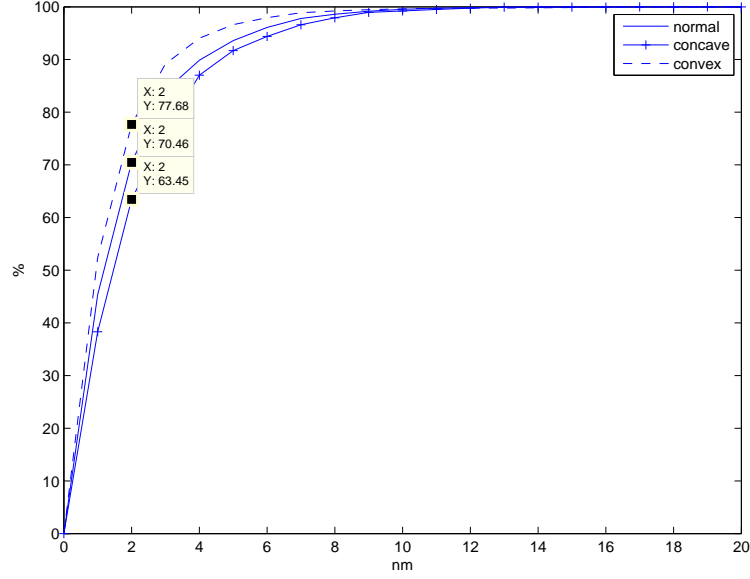


Fig. 8. Cumulative distribution of the absolute prediction error for a section of design A.

The RMSPE for design B is higher than design A, and the percentage of predicted movements in the right direction is lower for design B than for design A. Also, the percentage of absolute prediction errors below 2nm for design B is lower than of design A. We would expect the prediction on test data set from design A to outperform design B since the training data also comes from a portion of design A.

TABLE V

RMSPE and percentage of predicted edge movement in the same direction as model based OPC edge movement for three types of fragments. RMSPE is in nm. The data are from a section of layout from design B.

	# of edges	RMSPE	% Right Dir.
normal	4055	2.780	90.2
concave	455	3.935	90.8
convex	995	2.829	97.8

C. Improving convergence rate of model based OPC

We now provide the predictions as initial conditions to the iterations of model based OPC software to determine whether the number of iterations can be reduced. The predictions are applied before model

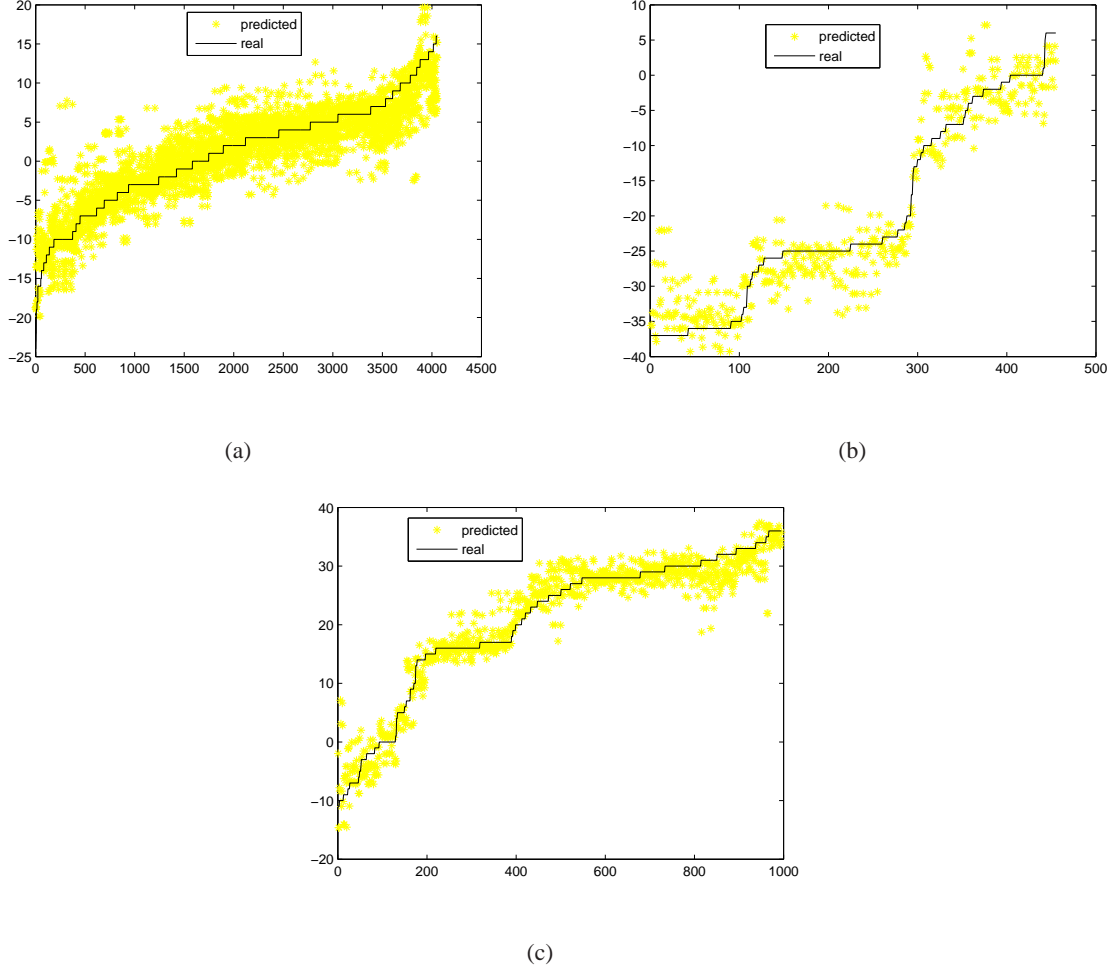


Fig. 9. The fragment movement obtained via model based OPC and the predicted fragment movement obtained via separate linear regression models for each type of fragment on a section of layout from design B. (a) normal edge fragment; (b) concave corner fragment; (c) convex corner fragment. The yellow indicates predicted movement from linear regression, and the black shows the movement obtained via model based OPC.

based OPC by creating tags on fragments for each value of predicted movement in Mentor Graphics Calibre™, and using the command “opcTag hintoffset”. Figure 11 shows the distributions of the edge placement errors (EPE) using model based OPC with and without the predictions from our proposed linear regression model for a section of layout from design A. As seen in Figure 11(a), the EPE with prediction after 2 iterations is more tightly distributed around zero than without predictions at 2 or 4 iterations. In fact, as shown in Figure 11(b) it takes 6 iterations for model based OPC without predictions to achieve approximately the same EPE distribution as 2 iterations of model based OPC with predictions.

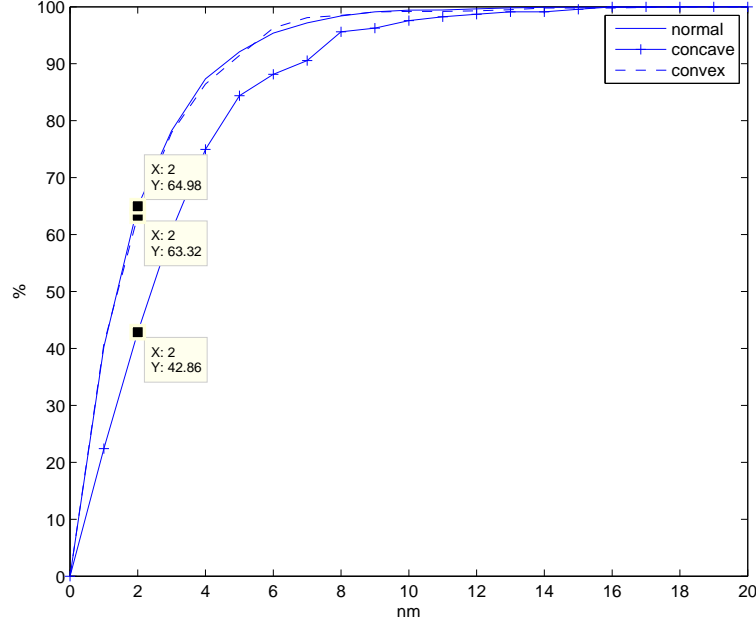


Fig. 10. Cumulative distribution of the absolute prediction error for a portion of layout from a design B.

Also notice that in Figure 11(b), even after 8 iterations without predictions, there is still a large number of EPEs between 4 and 5 nm. As shown in Figure 11(c), it takes 16 iterations of model based OPC without predictions to reduce the number of EPEs between 4 and 5 nm to the same level as 2 iterations of model based OPC with predictions. As seen in Table VI, the variance of the distribution for the iteration with prediction is approximately equal to that of 16 iterations without predictions. We can make similar observations for EPE distribution of design B test data set as shown in Figure 12 and Table VII. This reduction in iteration number corresponds to an overall reduction in the OPC run time. For instance, using the proposed prediction method, it takes 230 and 140 seconds to perform model based OPC for designs A and B respectively on a given computing platform. In comparison on the same computing platform, it takes 340 and 220 seconds to perform model based OPC without predictions for designs A and B to achieve the same EPE variance as the model based OPC with predictions. As such, in this example, OPC run time has been reduced by more than 32% when predictions via linear regression are applied.

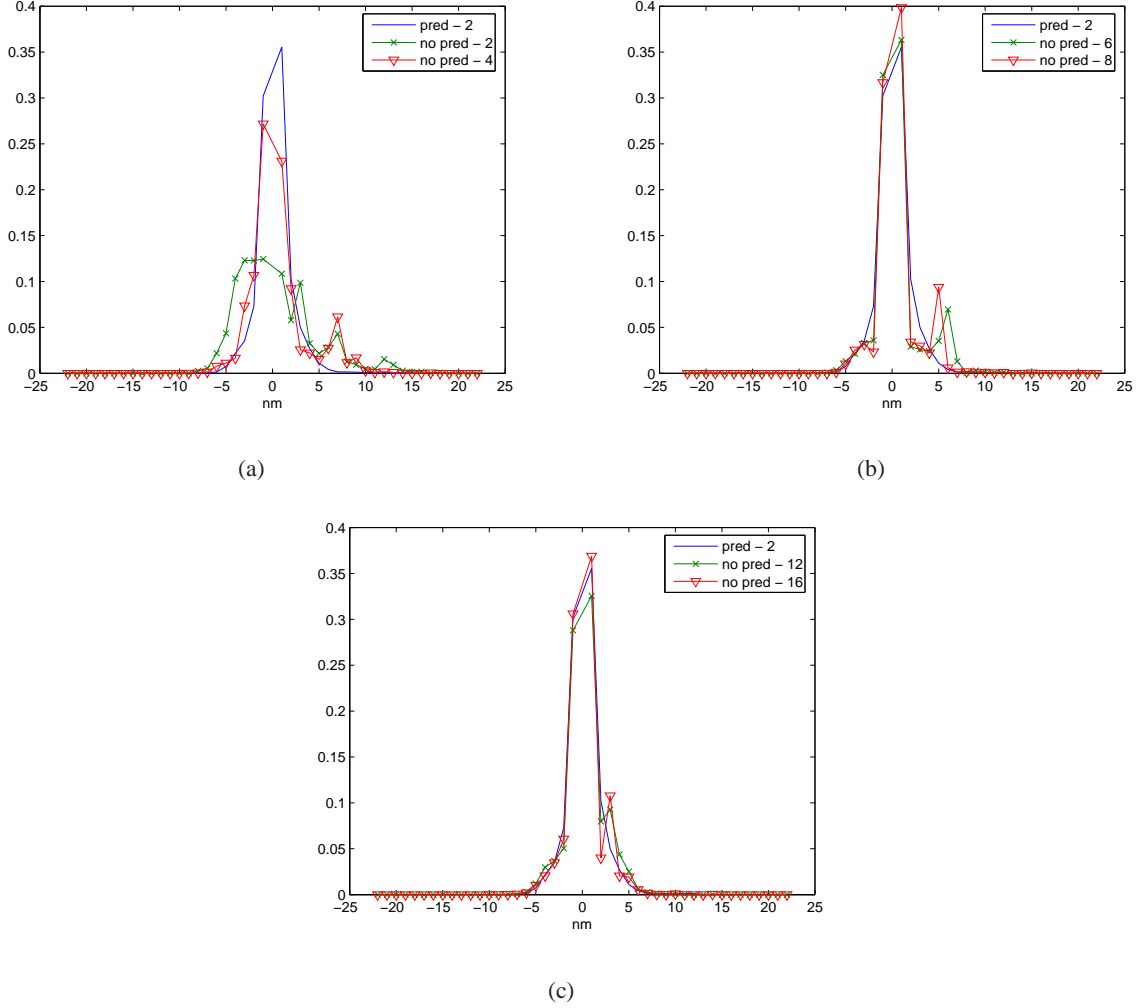


Fig. 11. The EPE distributions for 2, 4, 6, 8, 12, and 16 iterations of model based OPC without predictions and 2 iterations of model based OPC with predictions for design A. EPE is in nm. (a) 2 and 4 iterations of model based OPC without predictions and 2 iterations with predictions; (b) 6 and 8 iterations of model based OPC without predictions and 2 iterations with predictions; (c) 12 and 16 iterations of model based OPC without predictions and 2 iterations with predictions.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a method to predict OPC fragment movement from layout patterns using linear regression models. We use sub-sampled pixels on concentric squares of the low pass filtered layout pattern as input features, and create separate models for normal edge, concave corner, and convex corner fragments. We have shown that the proposed scheme can achieve RMSPE of 2.78 nm when the model is trained on one design and tested on a different design, and more than 63% of the absolute

TABLE VI

Variance and standard deviation for the various EPE distributions with and without predictions on design A.

	variance	std. dev
pred - 2	4.34	2.08
no pred - 2	18.95	4.35
no pred - 4	10.49	3.24
no pred - 6	7.06	2.66
no pred - 8	5.47	2.34
no pred - 12	5.03	2.24
no pred - 14	4.65	2.16
no pred - 16	4.28	2.07

TABLE VII

Variance and standard deviation for the various EPE distributions with and without predictions on design B.

	variance	std. dev.
pred - 2	6.81	2.61
no pred - 2	20.72	4.55
no pred - 4	11.70	3.42
no pred - 6	8.19	2.86
no pred - 8	6.53	2.56
no pred - 12	5.94	2.44
no pred - 4	5.49	2.34
no pred - 16	5.24	2.29

predictions errors for normal edge and convex corner fragments are less than 2 nm. In addition, using the predicted fragment movements presented in this paper as the initial condition for the iterations of Model based OPC, it is possible to reduce the number of iterations in model based OPC from 6 to 2 to obtain the same EPE distribution, and from 16 to 2 to obtain EPE distribution with approximately same variance. The reduction in the number of iterations translates into more than 32% reduction in the run time.

This paper has been primarily focused on reducing the overall computational time of model based OPC. There are other open issues in OPC which have not been addressed by the paper. For instance, it is well known that certain mask patterns such as "jogs" cause convergence problems in most model based OPC algorithms. Clearly, the work presented here is not readily applicable to the convergence problem.

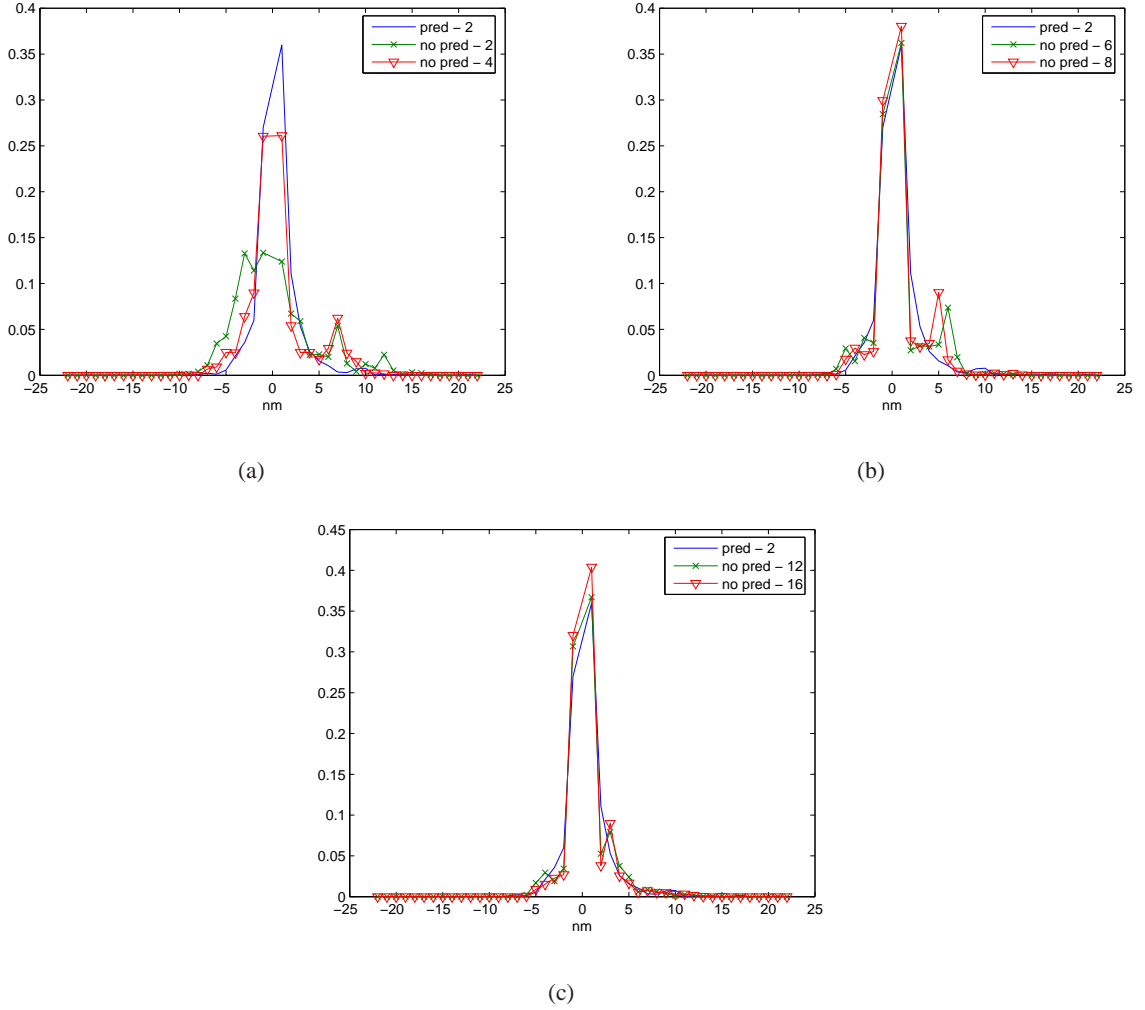


Fig. 12. The EPE distributions for 2, 4, 6, 8, 12, and 16 iterations of model based OPC without predictions and 2 iterations of model based OPC with predictions for design B. EPE is in nm. (a) 2 and 4 iterations of model based OPC without predictions and 2 iterations with predictions; (b) 6 and 8 iterations of model based OPC without predictions and 2 iterations with predictions; (c) 12 and 16 iterations of model based OPC without predictions and 2 iterations with predictions.

In the future, we plan to investigate more complex, general, distribution models, such as Gaussian Mixture model [21]. As shown in Section 3, more accurate predictions are obtained by creating a separate regression model for each type of fragment. However, it is considerably more difficult to separate the fragments within each fragment type in order to create separate regression models for improved predictions. This separation can potentially be learned from the data by using Gaussian mixture models.

Furthermore, all of the results presented in this paper have been for logic designs. In the future, we plan to investigate whether the same models can be applied to memory and mixed designs, and whether

the 32% reduction in OPC run time holds true across many designs.

REFERENCES

- [1] B. Saleh and S. Sayegh, "Reduction of errors of microphotographic reproductions by optimal corrections of original masks," Optical Engineering, vol. 20, pp. 781–784, 1981.
- [2] Y. Liu and A. Zakhor, "Binary and phase shifting mask design for optical lithography," Semiconductor Manufacturing, IEEE Transactions on, vol. 5, no. 2, pp. 138–152, 1992.
- [3] Y. Granik, "Fast pixel-based mask optimization for inverse lithography," Journal of Microlithography, Microfabrication, and Microsystems, vol. 5, p. 043002, 2006.
- [4] A. Poonawala and P. Milanfar, "Mask Design for Optical Microlithography—An Inverse Imaging Problem," Image Processing, IEEE Transactions on, vol. 16, no. 3, pp. 774–788, 2007.
- [5] J. Park, C. Park, S. Rhie, Y. Kim, M. Yoo, J. Kong, H. Kim, S. Yoo, S. Center, and S. CAE, "An efficient rule-based OPC approach using a DRC tool for 0.18 μ m ASIC," Quality Electronic Design, 2000. ISQED 2000. Proceedings. IEEE 2000 First International Symposium on, pp. 81–85, 2000.
- [6] J. Garofalo, K. Low, O. Otto, C. Pierrat, P. Vasudev, and C. Yuan, "Automatic proximity correction for 0.35 μ m I-line photolithography," Numerical Modeling of Processes and Devices for Integrated Circuits, 1994. NUPAD V., International Workshop on, pp. 92–94, 1994.
- [7] N. Cobb and A. Zakhor, "Fast sparse aerial-image calculation for OPC" in 15th Annual Symposium on Photomask Technology and Management, G.V. Shelden and J.N. Wiley, eds, Proc. SPIE 2621, pp. 534–545 (Dec. 1995), reprinted in Selected Papers on Resolution Enhancement Techniques in Optical Lithography, F.M. Schellenberg, ed., SPIE Milestone Series Vol. MS 178, SPIE Press, Bellingham, WA, 2004), pp. 805–816.
- [8] N. B. Cobb, A. Zakhor, and E. Miloslavsky, "Mathematical and CAD framework for proximity correction," in Proc. SPIE Vol. 2726, p. 208–222, Optical Microlithography IX, Gene E. Fuller; Ed., G. E. Fuller, Ed., vol. 2726, 1996, pp. 208–222.
- [9] N. Cobb, "Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing," Ph.D. dissertation, UNIVERSITY of CALIFORNIA, 1998.
- [10] 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., vol. 35, no. 12B Pt 1, pp. 6370–6373, 1996.
- [11] C. Park, S. Choi, S. Rhie, D. Kim, J. Park, T. Jang, J. Park, Y. Kim, M. Yoo, and J. Kong, "A hybrid PPC method based on the empirical etch model for the 0.14/spl mu/m DRAM generation and beyond," Quality Electronic Design, 2002. Proceedings. International Symposium on, pp. 143–147, 2002.
- [12] T. Kotani, S. Kobayashi, H. Ichikawa, S. Tanaka, S. Watanabe, and S. Inoue, "Advanced hybrid optical proximity correction system with OPC segment library and model-based correction module," in Proc. SPIE Vol. 4691, p. 188–195, Optical Microlithography XV, Anthony Yen; Ed., A. Yen, Ed., vol. 4691, Jul. 2002, pp. 188–195.
- [13] M. Hung and P. Balasingam, "Hybrid optical proximity correction: concepts and results," Proceedings of SPIE, vol. 4889, p. 1173, 2003.
- [14] R. Frye, E. Rietman, and K. Cummings, "Neural network proximity effect corrections for electron beam lithography," Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on, pp. 704–706, 1990.
- [15] P. Jedrasik, "Neural networks application for OPC (optical proximity correction) in mask making," Microelectronic Engineering, vol. 30, no. 1–4, 1996.
- [16] W. C. Huang, C. M. Lai, B. Luo, C. K. Tsai, M. H. Chih, C. W. Lai, C. C. Kuo, R. G. Liu, and H. T. Lin, "Intelligent model-based OPC," in Optical Microlithography XIX. Proceedings of the SPIE., vol. 6154, Apr. 2006, pp. 1065–1073.
- [17] N. Draper and H. Smith, Applied Regression Analysis. John Wiley & Sons New York, 1981.

- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004, p. 307.
- [19] *Selected Papers on Resolution Enhancement Techniques in Optical Lithography*, F. Schellenberg, Editor, SPIE Press, February 2004.
- [20] S. R. J. Brueck and X. Chen, “Spatial frequency analysis of optical lithography resolution enhancement techniques,” Journal of Vacuum Science Technology B: Microelectronics and Nanometer Structures, vol. 17, pp. 908–920, May 1999.
- [21] H. Sung, “Gaussian Mixture Regression and Classification,” Ph.D. dissertation, RICE UNIVERSITY, 2004.