

Optical Proximity Correction with Principal Component Regression

Peiran Gao, Allan Gu and Avideh Zakhori

Video and Image Processing Lab

Department of Electrical Engineering and Computer Science

University of California at Berkeley

{p_gao, 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. Then the mask is perturbed by moving the fragments to match the desired wafer pattern. 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; OPC for modern IC designs can take days to complete on computer clusters with thousands of CPU. In this paper, techniques from statistical machine 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 and efficient solution as the initial guess to model based OPC. To determine the best model, we train and evaluate several principal component regression models based on prediction error. Experimental results show that fragment movement predictions via regression model significantly decrease the number of iterations required in model based OPC.

Keywords: principal component regression, statistical learning, OPC, IC Layout

1. INTRODUCTION

An important step in today's IC manufacturing is optical proximity correction (OPC); today, 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 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; OPC can take days to complete on computer clusters with thousands of CPU for modern IC designs.

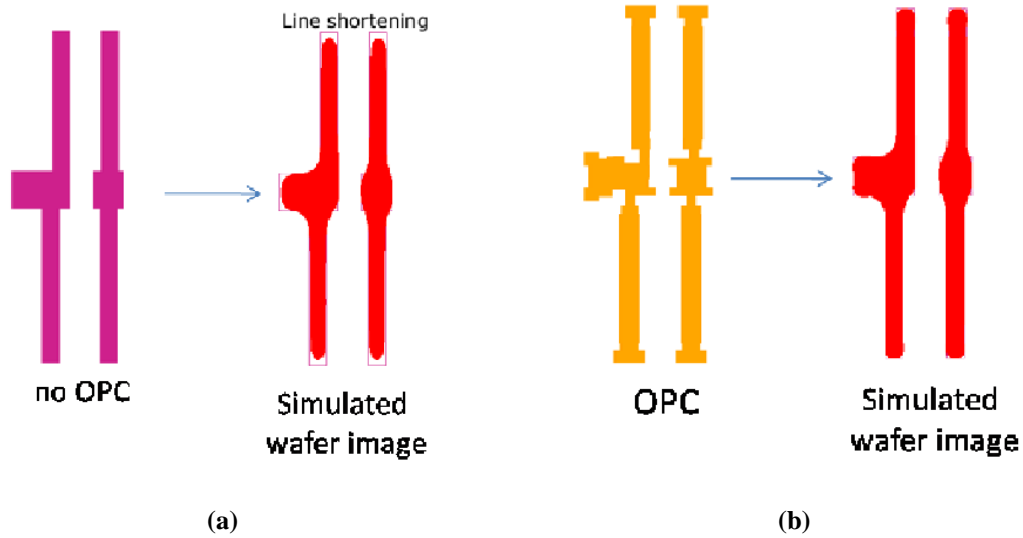


Figure 1: (a) Layout pattern without OPC and the simulated image on the wafer; (b) layout pattern with OPC and the simulated image on the wafer.

There are two types of 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^{1,2}. Unlike rule based OPC, model based OPC^{3,4} typically uses simulations in an iterative manner. The layout consisting of polygons 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 computationally intensive due to the simulations that are performed at each iteration. 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, can reduce the overall OPC run time and product development cycle time. In practice, a hybrid solution is usually taken by OPC engineers with simple rule based OPC being followed by model based OPC^{5,6,7,8}.

In this paper, we propose a preprocessing step for model based OPC, using regression techniques from machine learning so as to decrease the number of iterations required for convergence. Our previous work using ordinary linear regression has shown increased rate of convergence for model based OPC;⁹ however, we have also observed linear regression to be unstable, and to result in large errors for certain layouts such as memory. There has been prior work on using neural networks to correct for non-idealities in lithography systems. Frye *et al.*¹⁰ 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. Jedrasik¹¹ has proposed a neural network approach for one step OPC. Recently, Huang *et al.*¹² 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: In Section 2, we present principal component regression (PCR) as a way to predict the movement of layout fragments, and compare it with ordinary linear regression. Section 3 describes the training and evaluation methodology used to select the best PCR model. Section 4 presents the prediction results on different portions of two 90nm layouts using the PCR model selected in Section 3. In Section 4, we show that the fragment movements obtained via PCR can be used as initial condition for model based OPC to reduce the number of iterations. Finally, conclusions and future work are presented in Section 5.

2. REGRESSION TECHNIQUES

2.1 Ordinary Linear Regression

Linear regression is a statistical technique 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 shown¹³ that

$$\vec{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \vec{y}) \quad (2)$$

is the solution to the least square problem in Equation 1.

2.2 Principal Component Regression (PCR)

Due to non-ideal conditions, such as repetitive patterns in the memory section of a layout, the matrix $\mathbf{X}^T \mathbf{X}$ can sometimes become ill-conditioned or even singular so that its inverse cannot be accurately computed. An example of linear regression's numerical instability for this application is shown later in Section 4. As such, we introduce a modified version of ordinary linear regression, namely PCR, to address this problem.

PCR is a statistical technique that reduces the dimensionality of the input feature vector using principal component analysis before modeling the dependence of the output y on input \vec{x} ^{15,16}. The idea is to eliminate redundant components of the input feature vector before applying regression. To accomplish this, we first diagonalize the covariance matrix: $\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$, where columns of \mathbf{V} are principal components, and $\mathbf{\Lambda}$ is a diagonal matrix corresponding to variances of the principle components. We assume the principal components with higher variances to have a larger influence on the prediction of the model. After ranking the principal components by their variances, we use the top $n\%$ of principal components as column vectors to form a transformation matrix \mathbf{T} to eliminate redundancy in feature vectors. In our experiments, the optimal value of n is determined empirically to be 90. The new regression model becomes $y = \vec{\beta}^T (\vec{x} \mathbf{T}) + \epsilon$.

3. TRAINING AND EVALUATION

In this section, we determine the best input feature and its associated dimension. Mentor Graphics Calibre™ is used to perform model based OPC using a vector optical model with wavelength $\lambda = 193\text{nm}$ and $NA = 0.85$ for two 90nm IC designs. An annular aperture with $\sigma = 0.88, 0.44$ and a VT5 resist model is used. The surrounding $2\mu\text{m} \times 2\mu\text{m}$ layout pattern for each fragment is captured and used to derive its movement using our proposed method. We choose the $2\mu\text{m} \times 2\mu\text{m}$ 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 \times 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 $2\mu\text{m} \times 2\mu\text{m}$ 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. Gu and Zakhor⁹ have shown that training a separate model for each fragment type results in improved performance.

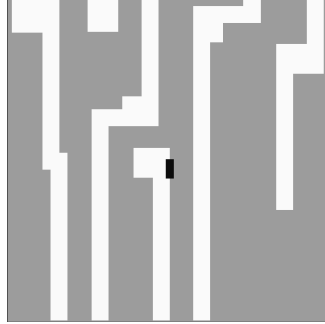


Figure 2: Example of $2\mu\text{m} \times 2\mu\text{m}$ layout pattern. The fragment of interest is in black.

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 can at most increase the maximum spatial frequency to $\frac{2NA}{\lambda}$.¹⁴ As such, we choose to use low pass filter layout patterns as features in our PCR model to predict the fragment movements. Specifically, we filter the

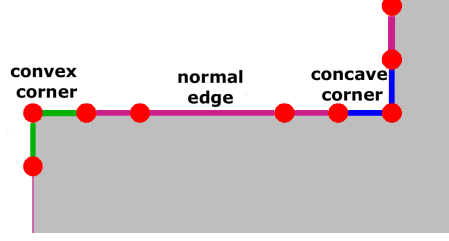


Figure 3: Examples of normal edge, convex corner, and concave corner fragments. Normal edge fragments are in violet, convex corner fragments are in green, and concave corner fragments are in blue.

$2\mu\text{m} \times 2\mu\text{m}$ 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.

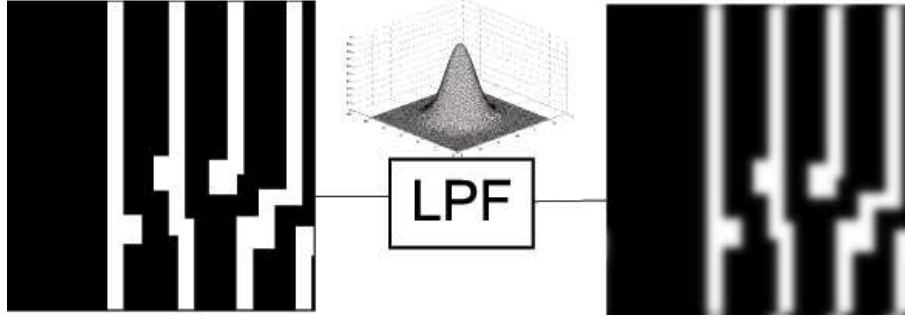


Figure 4: Original layout pattern and the resulting filtered pattern.

The optimal fragment movement depends on the $2\mu\text{m} \times 2\mu\text{m}$ pattern. However, it is impractical to use all the layout pixels as the input feature vector; specifically, the $400 \text{ pixel} \times 400 \text{ pixel}$ pattern translates into a feature vector in \mathbb{R}^{160000} . For this high dimensional vector, evaluating the value of $\vec{\beta}$ in the principal component regression model is quite computationally intensive.

Since OPC is essentially an optical correction process, the effects of a mask pattern on the wafer obey the inverse square law; this means that the intensity of light from a source is inversely proportional to the square of the distance from the source.¹⁷ In other words, the effect of mask patterns on the fragment being moved on the wafer during OPC is inversely proportional to the square of the distance between a fragment and the corresponding mask pattern.

As such, we propose to use an incremental concentric square sampling (ICSS) method as shown in Figure 5 to create feature vectors. Concentric squares centered at the middle of the 400×400 pixel pattern are super-

imposed on top of the pattern. In order to follow the quadratic pattern implied by the inverse square law, the side length L_i for the i th square is recursively defined as $L_i = L_{i-1} + 2\alpha i$, where $L_0 = 0$. For $\alpha = 1$, the first few side lengths are 0, 2, 6, 12, 20 pixels. Pixel values are sampled at the 4 corners and the mid point of each side of each concentric square to form the feature vector. Note that α has to be an integer in order to avoid non-integer coordinates or oversampling near the center. We have empirically shown ICSS to outperform the concentric square sampling method used by Gu and Zakhori;⁹ this is because it increases the sampling density close to the fragment, and decreases the sampling density further away from the fragment while reducing the feature vector size.

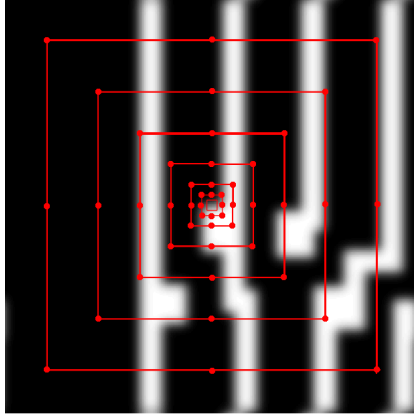


Figure 5: *Incremental Concentric Square Sampling of Pixel Values.*

In the remainder of this section, experiments are performed to determine the optimal value of α . In Section 4, 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

$$RMSPE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y})^2} \quad (3)$$

where y_i is the fragment movement determined by model based OPC, and \hat{y} is the predicted fragment movement obtained in our proposed principal component regression model. The smaller the RMSPE value, the more effective is the model at predicting the fragment movement. In this paper, the RMSPE is calculated on new test data sets that are not used during the training process.

3.1 Training

We perform model based OPC on a $50\mu\text{m} \times 50\mu\text{m}$ section of logic layout from design A for 30 iterations, and select 4562 pairs of fragment movements and associated patterns among all the available pairs in the $50\mu\text{m} \times 50\mu\text{m}$ 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 $50\mu\text{m} \times 50\mu\text{m}$ patch, we make sure that at least one of the 4562 pairs correspond to each fragment movements in the range of movements predicted by model based OPC. About half of the 4562 pairs, or 2318 pairs are used for training, and the remaining half are used for testing. The 4562 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 2318 training pairs consist of 854 pairs with normal edge, 890 pairs with convex corner, and 574 pairs with concave corner fragments.

Separate PCR models are trained for each type of fragment with fragment movements and feature vectors obtained by applying ICSS on associated patterns with $\alpha = \{1, 2, 3, 4, 5\}$. The resulting sizes of feature vectors are 161, 121, 97, 89, 81.

3.2 Evaluation

As described earlier, we use 2244 pairs from design A for evaluation. The testing set consists of 839 pairs with normal edge, 861 pairs with convex corner, and 544 pairs with concave corner fragments. The PCR models are trained with different values for α from $\alpha = 1$ to $\alpha = 5$. For each pattern, the fragment movement is predicted as $\hat{y} = \vec{\beta}^T(\vec{x}T)$, where \vec{x} is a feature vector containing 3 indicator variables and the sampled pixel values, and T is the transformation matrix obtained from principal component analysis. The RMSPEs as defined in Equation 3 for different values of α are shown in Table 1 for models generated for each type of fragments separately. The last row shows the overall RMSPE of all three types using separate models for each type. As seen, $\alpha = 5$ and $\alpha = 1$ result in the highest and lowest overall RMSPE of all three fragment types at 3.398nm and 3.190nm respectively. This is expected, since $\alpha = 1$ corresponds to the highest sampling density. Even though $\alpha = 2$ results in slightly lower RMSPEs for convex and concave fragments, $\alpha = 1$ outperforms it with a much lower RMSPE for normal fragments. As such, we choose to use a separate model for each fragment type using ICSS

with $\alpha = 1$.

	Number of fragments		RMSPE				
	Training	Evaluation	$\alpha = 1$ (161)	$\alpha = 2$ (121)	$\alpha = 3$ (97)	$\alpha = 4$ (89)	$\alpha = 5$ (81)
normal	854	839	3.264	3.469	3.401	3.584	3.579
convex	890	861	2.904	2.877	2.945	3.034	3.069
concave	574	544	3.492	3.491	3.415	3.609	3.600
overall	2318	2244	3.190	3.260	3.237	3.390	3.398

Table 1: RMSPE for different values of α for different fragment types separated and combined. RMSPE is in nm. In parenthesis are the sizes of feature vectors corresponding to α .

4. RESULTS

In this section, we use ICSS with $\alpha = 1$ as input feature vector to train separate models for normal edge, concave corner, and convex corner fragments for a $50\mu\text{m} \times 50\mu\text{m}$ portion of design A used in Section 3. Even though the model obtained in Section 3 also corresponds to ICSS with $\alpha = 1$, its training set does not necessarily cover the entire range of fragment movements for different fragment types. To this end, we collect a subset of all the pairs of layout patterns and their corresponding fragment movements obtained via 30 iterations of model based OPC for the $50\mu\text{m} \times 50\mu\text{m}$ patch of layout A for training purpose. Specifically, we collect 3443 pairs of patterns with normal edge, 2926 pairs with convex corner, and 2345 pairs with concave 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, $\vec{\beta}$, and transformation matrix, T , are used to test on three different sections of layout from two different designs.

4.1 Comparison with model based OPC

We test the model on three different sections of layout from two different designs. The first one is a $90\mu\text{m} \times 80\mu\text{m}$ section of logic layout from design A with 68246 normal edges, 20380 convex corner, and 7637 concave corner fragments. The second one is a $30\mu\text{m} \times 30\mu\text{m}$ section of logic layout from design B with 9893 normal edges, 2674 convex corner, and 893 concave corner fragments. Finally the third one is a $45\mu\text{m} \times 45\mu\text{m}$ section of mixed logic and memory layout from design B with 26790 normal edges, 10193 convex corner, and 2345

concave corner fragments. We name the three layout sections Logic A, Logic B, and Mixed B respectively for simplicity. Table 2 shows the RMSPE for different fragment types and the percentage of times the direction of movement predicted by PCR matches that of model based OPC. As seen, our models result in the lowest RMSPE for normal edges, with more than 88% of predicted fragment movements for all three layout sections being in the same direction as that of model based OPC. On the other hand, our models achieve the highest RMSPE for concave corner fragments, with more than 75%, 89% and 59% of predicted fragment movements being in the same direction as that of model based OPC for Logic A, Logic B, and Mixed B respectively. This can partially be attributed to fewer training examples for concave corners as shown in Table 1, possibly resulting in a less accurate model.

	Logic A			Logic B			Mixed B		
	Frag. #	RMSPE	% Right Dir.	Frag. #	RMSPE	% Right Dir.	Frag. #	RMSPE	% Right Dir.
normal	68246	4.717	90.5	9893	3.335	88.2	26790	3.804	91.0
convex	20380	5.873	89.5	2674	3.751	96.8	10193	7.134	86.4
concave	7637	8.213	75.1	893	5.730	89.6	2345	8.927	59.5

Table 2: Number of fragments, 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 Logic A, Logic B, and Mixed B.

As shown in Figures 6, 7 and 8, the predictions via PCR models shown in yellow follow the fragment movements obtained via 30 iterations of model based OPC in black reasonably well for all three layout sections. The direction of movement for normal edges is approximately centered around zero, while those of convex and concave corners are centered around positive and negative values respectively. This is expected because on average we would expect convex corners to move outward and concave corner to move inward. Also, visual inspection of the concave corner plots for all three layout sections reveals a lower performance as compared to normal edge or convex corner. This is in agreement with RMSPE results in Table 2.

Figure 9 shows the cumulative distribution function (CDF) of the absolute prediction error for all three layout sections. Averaged across all layout sections, around 80%, 70% and 50% of the absolute prediction errors are less than 5nm for normal edges, convex corners and concave corners respectively. The relatively

lower performance on concave corner in Figure 9 is in agreement with Table 2 and Figures 6, 7, and 8.

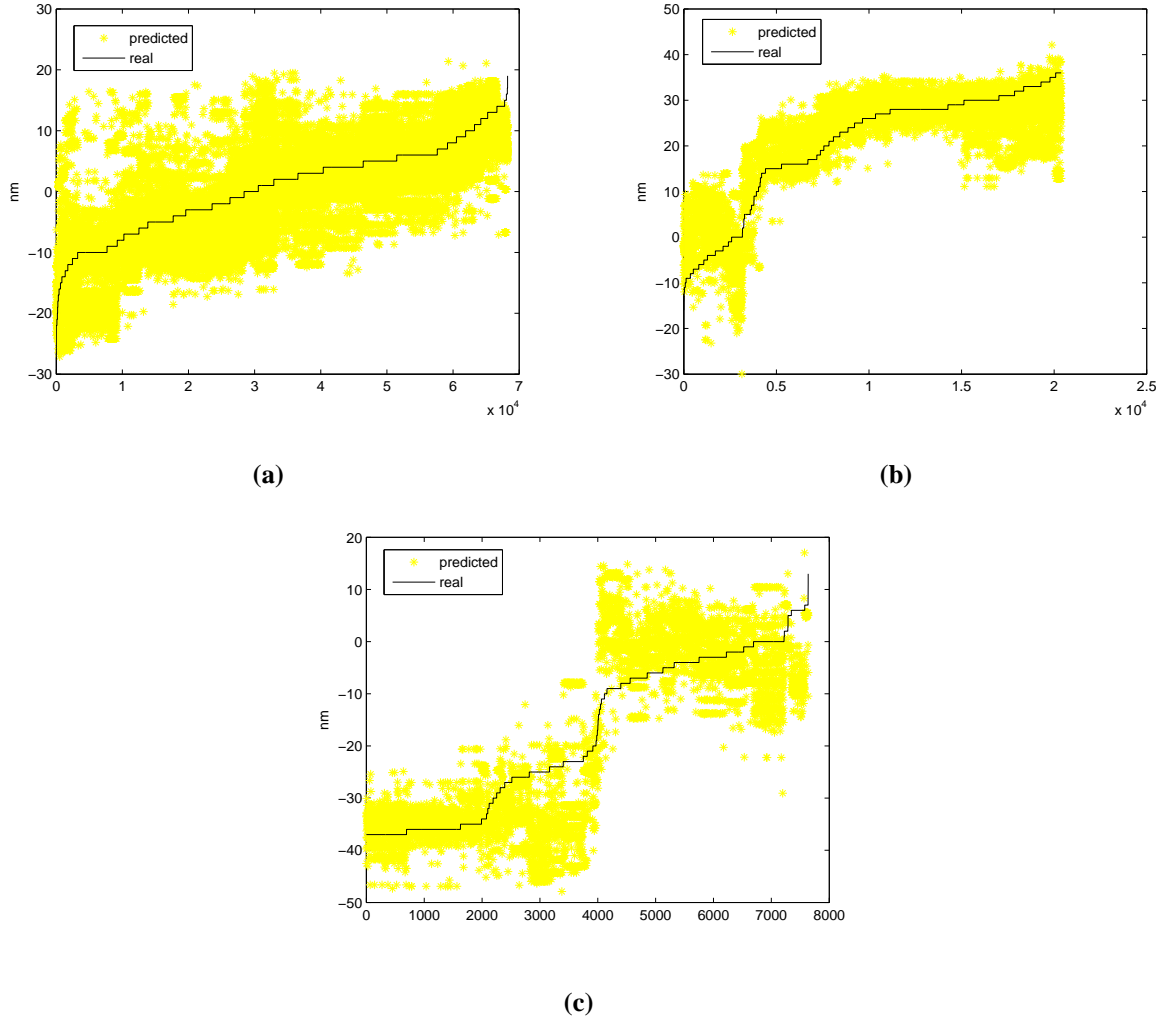


Figure 6: The model based OPC fragment movement and the predicted fragment movement using separate PCR models for each fragment type on Logic A. (a) normal edge; (b) convex corner; (c) concave corner. The yellow indicates predicted movement using PCR, and the black shows the movement obtained from model based OPC software.

For all three fragment types, the RMSPE, percentage of right direction, and the CDF show better performance of our models for Logic B than for Logic A. This is somewhat unexpected since the training set actually comes from design A. However, upon closer inspection, Logic B shares similar pattern density with the training set, while the pattern density of Logic A is higher than that of the training set. This implies that our PCR model is inherently sensitive to the pattern density of the layout on which it is trained. Our model's performance on

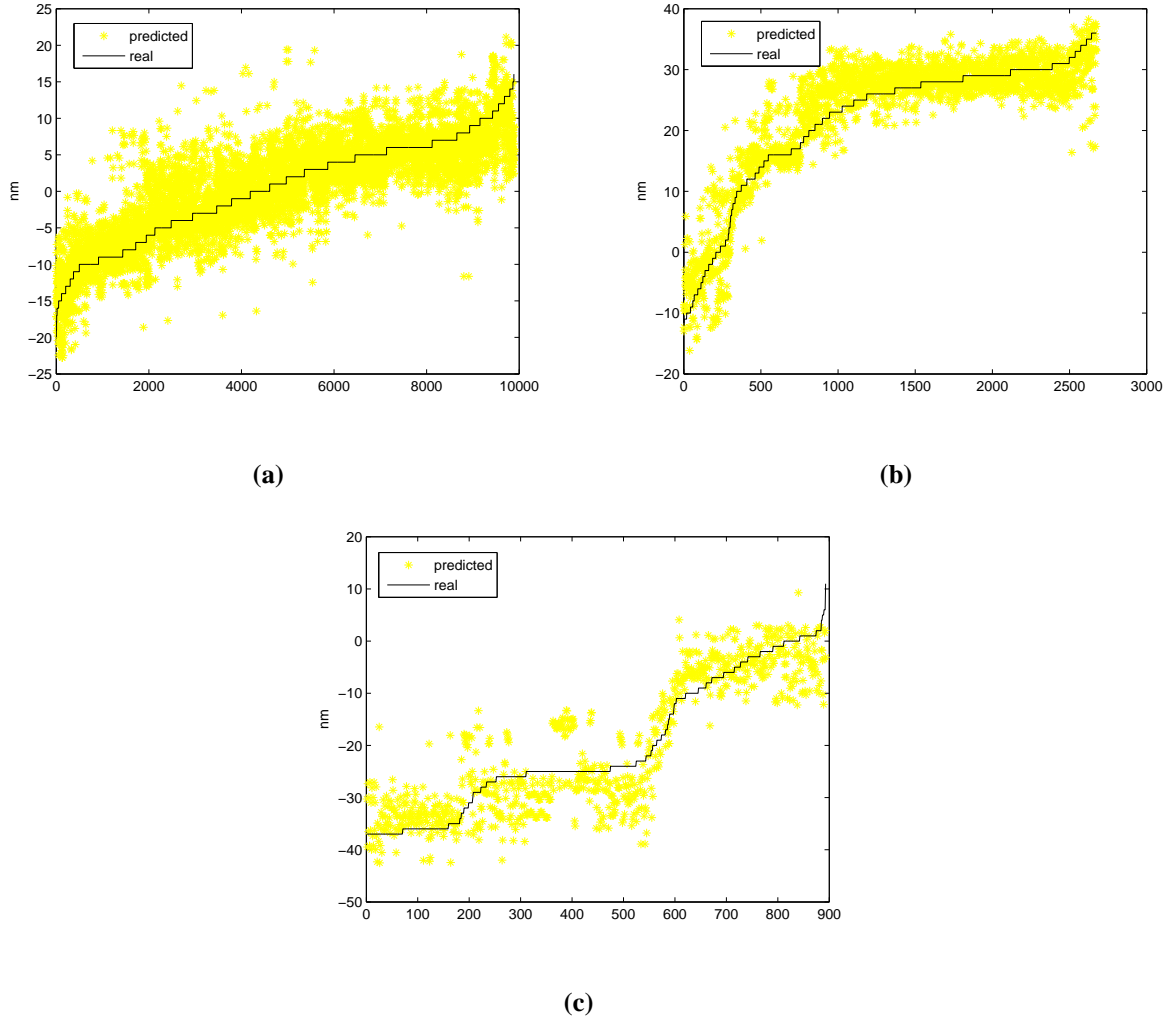


Figure 7: The model based OPC fragment movement and the predicted fragment movement using separate PCR models for each fragment type on Logic B. (a) normal edge; (b) convex corner; (c) concave corner. The yellow indicates predicted movement using PCR, and the black shows the movement obtained from model based OPC software.

Mixed B also shows that our PCR model is compatible with both memory and logic layouts.

4.2 Improving convergence rate of model based OPC

We now provide the predictions obtained from our PCR models as initial conditions to the iterations of model based OPC software to determine the extent to which the number of iterations can be reduced. The predictions are applied before model based OPC by creating tags on fragments for each value of predicted movement in Mentor Graphics Calibre™, and using the command “opcTag hintoffset”. The results of model based OPC

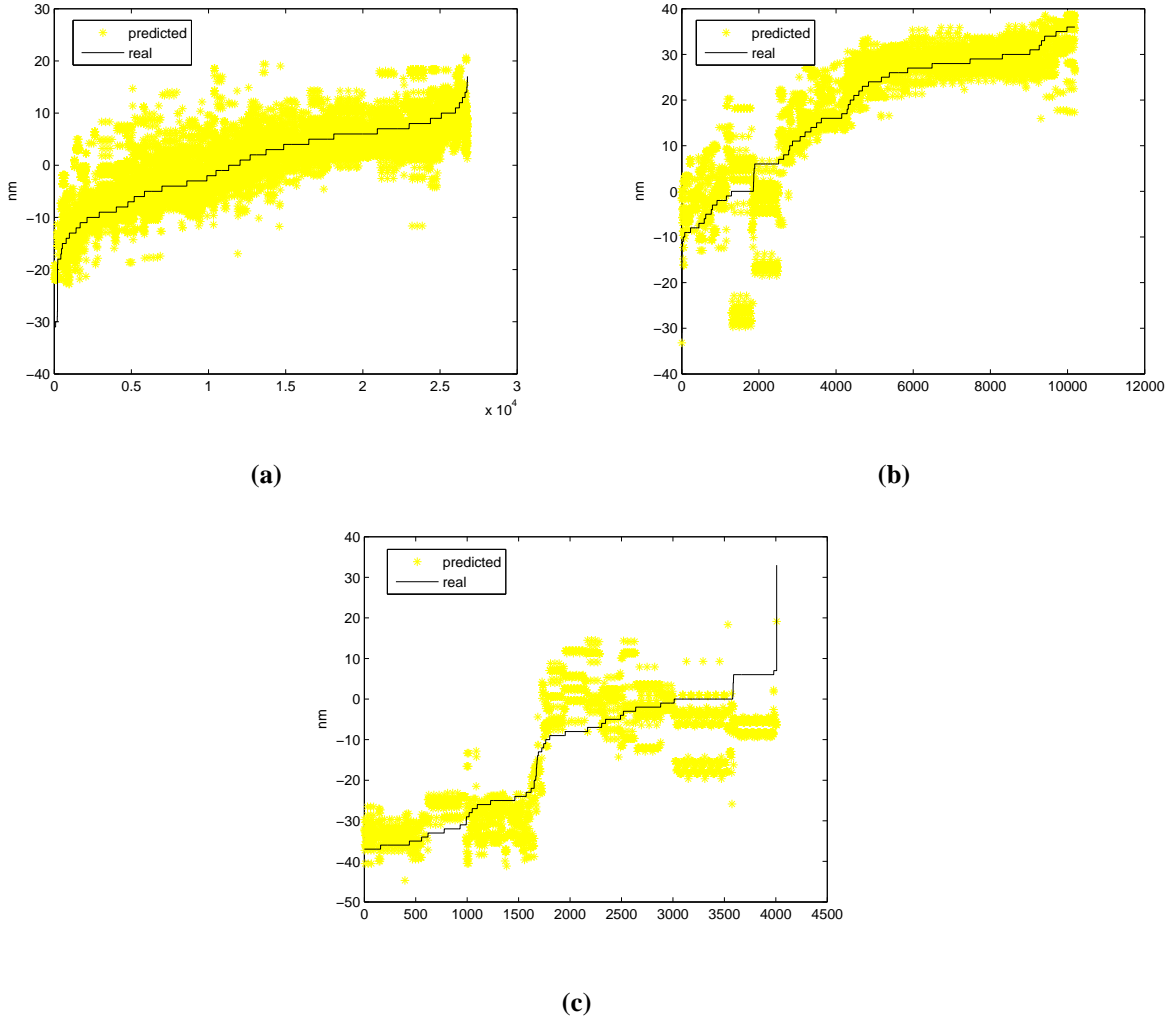


Figure 8: The model based OPC fragment movement and the predicted fragment movement using separate PCR models for each fragment type on Mixed B. (a) normal edge; (b) convex corner; (c) concave corner. The yellow indicates predicted movement using PCR, and the black shows the movement obtained from model based OPC software.

are compared using edge placement errors (EPE), the distance between actual edge position on the wafer and the desired position. Figure 10 shows the distributions of EPE using model based OPC with and without the predictions from our proposed PCR model for Logic A, Logic B, and Mixed B. As seen in Figure 10, in all three cases, the EPE with prediction after 2 iterations is much more tightly distributed around zero than without predictions at 2 iterations. In fact, after 2 iterations with predictions, the EPE distributions are close to those after 8 iterations without predictions. Also notice that in Figure 10, even after 8 iterations without predictions,

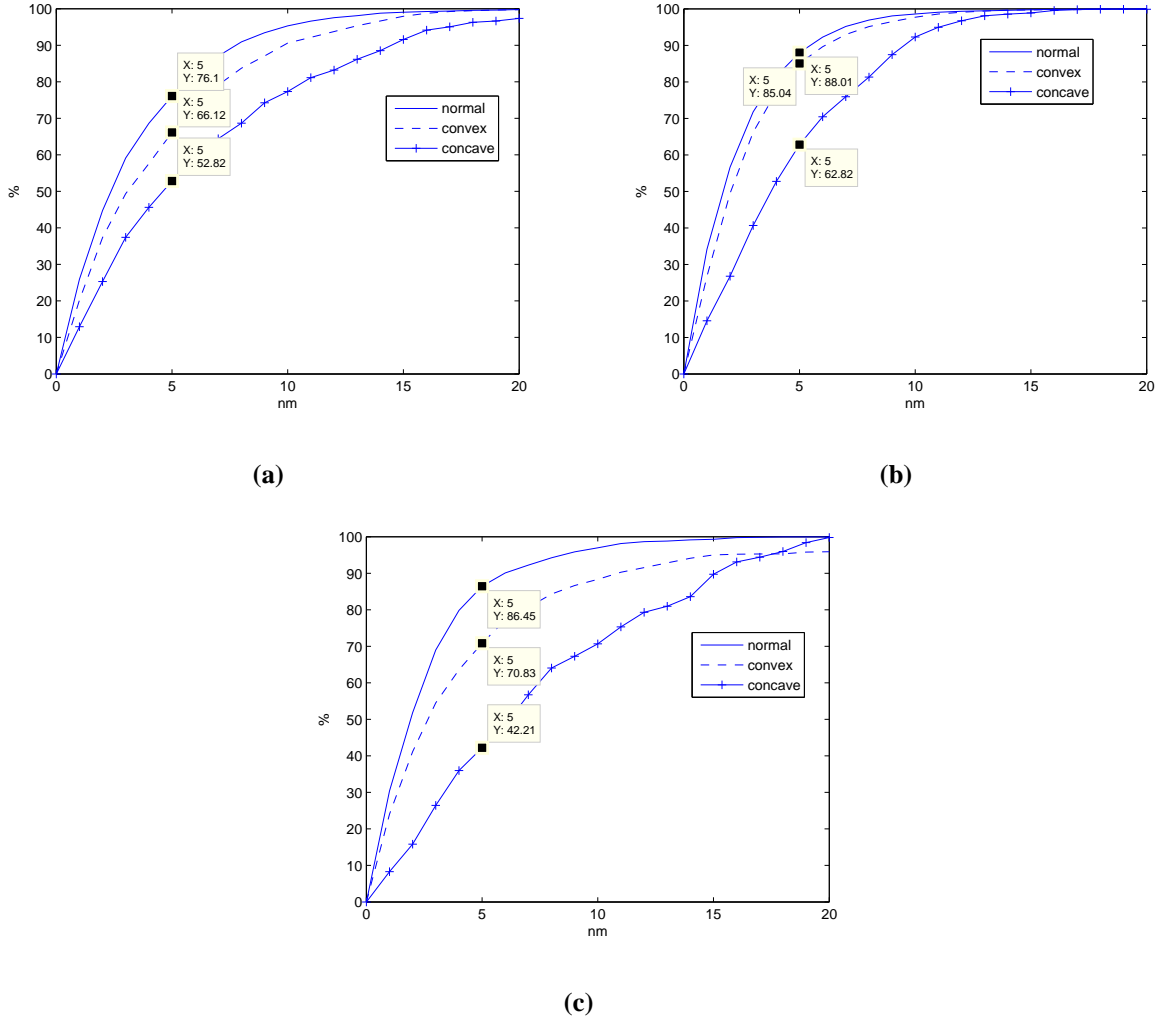


Figure 9: Cumulative distribution of the absolute prediction error for Logic A, Logic B, and Mixed B.

there is still a large number of EPEs in the distribution between 4nm and 5 nm, which is absent in the cases with predictions. The standard deviations of EPE distributions for all three layout sections are summarized in Table 3. The standard deviation of the distribution after 2 iterations with predictions using methods described in this paper is approximately equal that of 6 to 8 iterations without predictions for the three cases, indicating a reduction in model based OPC by 6 to 8 iterations. Considering that all tested sections converge within 30 iterations of model based OPC, this corresponds to a minimum of 20% to 27% of savings in runtime of model based OPC.

Predictions made using Gu and Zakhor's linear regression technique⁹ on the same sections of layouts are

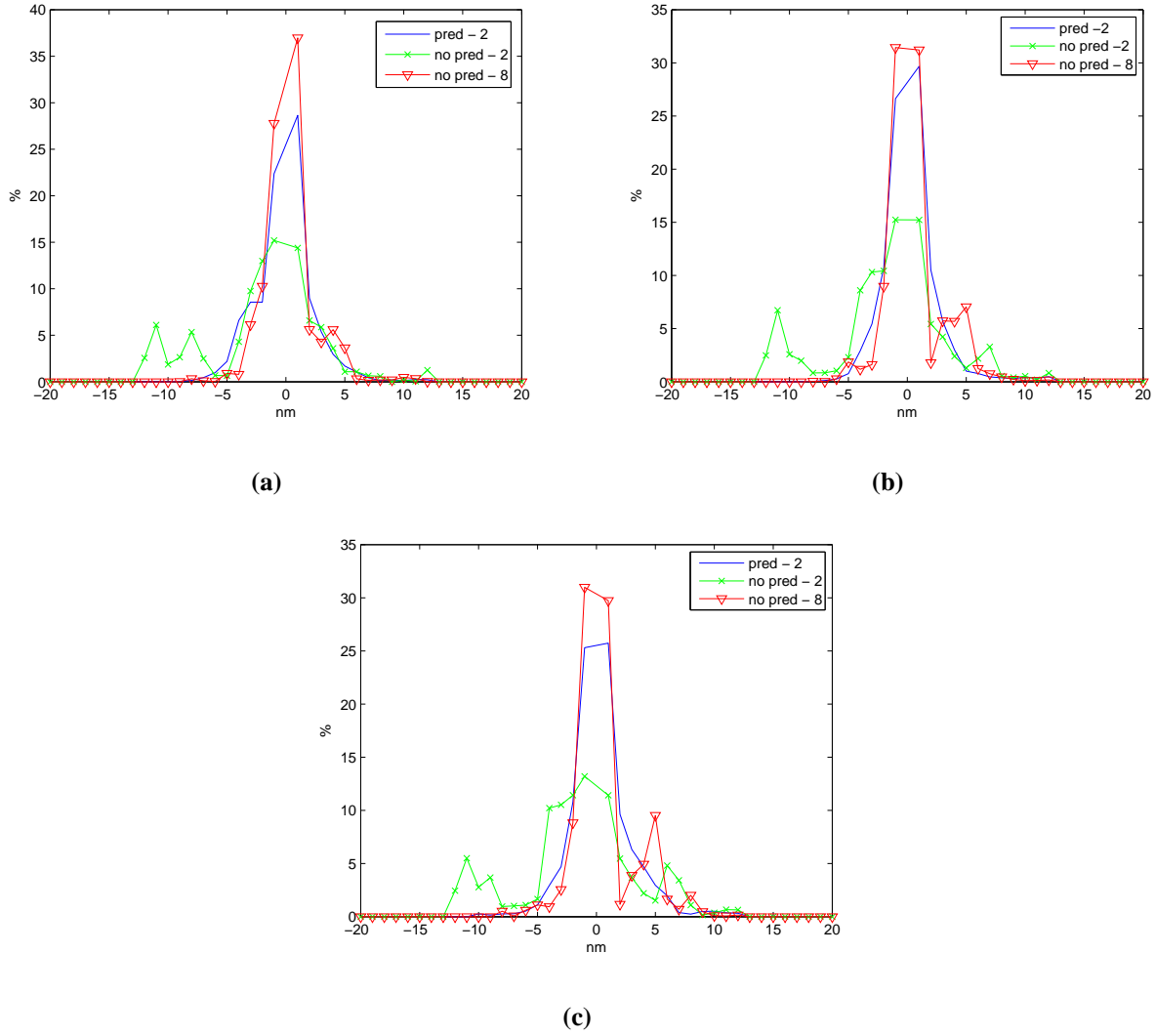


Figure 10: The EPE distributions for 2 and 8 iterations of model based OPC without predictions and 2 iterations of model based OPC with predictions. EPE is in nm. (a) 2 and 8 iterations of model based OPC without predictions and 2 iterations with predictions for Logic A; (b) 2 and 8 iterations of model based OPC without predictions and 2 iterations with predictions for Logic B; (c) 2 and 8 iterations of model based OPC without predictions and 2 iterations with predictions for Mixed B.

also shown in the 3rd row of Table 3. As seen PCR results in lower EPE standard deviation in two out of three tested layout sections. The advantage of PCR over linear regression, though, is improved stability. For instance, as shown in Table 4, when both linear regression and PCR models are trained and tested on sections of memory layout from design A with a large amount of repetitive patterns, the RMSPE of linear regression models becomes

	Logic A	Logic B	Mixed B
PCR, pred - 2	2.31	2.18	2.50
Linear regression, ⁹ pred - 2	1.98	2.34	2.71
no pred - 2	4.51	4.61	4.78
no pred - 4	3.14	3.28	3.55
no pred - 6	2.44	2.61	2.92
no pred - 8	2.03	2.21	2.51
no pred - 10	1.85	2.05	2.31
# of iterations saved by PCR	> 6	> 8	> 8

Table 3: Standard deviation for the various EPE distributions with and without predictions.

extremely large as the result of numerical instability, while the RMSPE of PCR models is comparable to that of other experiments conducted in this paper.

Fragment type	# of edges	RMSPE	
		Linear Regression	PCR
Normal	3968	107,412.3	1.1
Convex	3596	3,917,175.2	4.5
Concave	1888	13.5	3.4

Table 4: RMSPE for a section of memory layout on design A tested with with linear regression and PCR. RMSPE is in nm.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a technique for predicting OPC fragment movement from layout patterns using PCR models. We use incremental concentric square sub-sampled pixels 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 ICSS and PCR combination can achieve RMSPE of between 3nm and 9nm when the model is trained on a part of design A and tested on sections from both design A and design B. 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 8 to 2 to obtain similar EPE distribution standard deviation. This corresponds to time saving of 20-27% in the runtime of the OPC step, and therefore shortens product development cycle time. In addition, we have shown that one set of PCR models is capable of predicting fragment movements of different layout patterns from multiple designs with the same minimum feature size.

In the future, we plan to investigate more complex distribution models, such as a Gaussian Mixture models,¹⁸ to further increase the accuracy of our model's predictions.

6. ACKNOWLEDGEMENT

This research was conducted under the Research Network for Advanced Lithography, supported jointly by the Semiconductor Research Corporation 2005-OC-460 and the Defense Advanced Research Project Agency W911NF-04-1-0304. We would also like to acknowledge Brian Richards of the University of California, Berkeley for his help in setting up the model based OPC engine used in this paper.

REFERENCES

1. 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.
2. J. Garofalo, K. Low, O. Otto, C. Pierrat, P. Vasudev, and C. Yuan, "Automatic proximity correction for 0.35 μ m I-linphotolithography," *Numerical Modeling of Processes and Devices for Integrated Circuits, 1994. NUPAD V, International Workshop on*, pp. 92-94, 1994.
3. 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.
4. N. Cobb, *Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing*. PhD thesis, UNIVERSITY of CALIFORNIA, 1998.
5. 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.

6. 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 Eletronic Design, 2002. Proceedings. International Symposium on*, pp. 143-147, 2002.
7. 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.
8. M. Hung and P. Balasingam, "Hybrid optical proximity correction: concepts and results," *Proceedings of SPIE*, vol. 4889, p. 1173, 2003.
9. A. Gu and A. Zakhor, "Optical Proximity Correction with Linear Regression," *IEEE Transactions on Semiconductor Manufacturing*, Accepted for publication November 2007.
10. 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.
11. P. Jedrasik, "Neural networks application for OPC (optical proximity correction) in mask making," *Microelectronic Engineering* **30**(1-4), 1996.
12. 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.*, **6154**, pp. 1065-1073, Apr. 2006.
13. S. Boyd and L. Vandenberghe, *Convex Optimization*, p. 307. Cambridge University Press, 2004.
14. 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* **17**, pp. 908-920, May 1999.
15. G. B. Wetherill, *Regression Analysis with Applications*, p. 94. Capman and Hall, 1986.
16. J. E. Jackson, *A User's Guide To Principal Components*, p. 271. Wiley-Interscience Publication, 1991.
17. E. Hecht, *Optics*, p. 56. Addison Wesley, 2001.
18. H. Sung, *Gaussian Mixture Regression and Classification*. PhD thesis, RICE UNIVERSITY, 2004.