A New Intensity Based Edge Placement Error Optimization Algorithm for Optical Lithography

Ahmed Awad*, Atsushi Takahashi*, Satoshi Tanaka†, and Chikaaki Kodama†
*Department of Communications and Computer Engineering, Tokyo Institute of Technology, Tokyo, Japan
Email: {awad.ahmed, atsushi}@eda.ce.titech.ac.jp
†Toshiba Corporation, Kanagawa, Japan
Email: {satoshi2.tanaka, chikaaki1.kodama}@toshiba.co.jp

***Abstract*—With the continuous shrinking of minimum feature sizes beyond current 193nm wavelength for optical micro lithography, the electronic industry relies on Resolution Enhancement Techniques (RETs) to improve pattern transfer fidelity. Dense layouts suffer from more and more image quality degradation that badly impacts lithographic yield. In this paper, a new intensity based algorithm is proposed to minimize Edge Placement Error (EPE) in low scale features printing through simultaneous shifting of neighboring segments and serifis sizing for corners. Our experimental results show that our algorithm minimizes EPE effectively on the public benchmarks released by IBM for ICCAD 2013 contest.**

I. Introduction

Optical lithography, wherein, an integrated circuit (IC) is patterned layer by layer, is the cardinal part of IC fabrication process. In the design level, electrical structures are defined as a set of polygons that are carved by a pixelated template called mask [1]. An optical system (usually called as an exposure tool) is used to project the image of mask patterns to a silicon wafer which is coated by photoresist (photo sensitive polymer). If the exposure dose is greater than a certain threshold level, the exposed region of the resist will be chemically changed and formed the patterns under the development process, and finally etched to transfer to the wafer. To keep pace the increased shrinking in features sizes, optical wavelength of an exposure tool had been steadily reduced from 436nm (Hg lamp) to 193nm (ArF excimer laser), and it reached its practical limit such as high instability and strong birefringence of lens materials [2]. To obtain more small features printability, immersion lithography was introduced to increase the Numerical Aperture (NA). However, sub-100nm nodes printing suffers from image distortions which badly impact the proper functionality of the circuit. This gave the birth of Optical Proximity Correction (OPC), in which the mask shape is modified to improve the image quality [3].

Many lithographic models were introduced to simulate the printed image of a mask on the wafer in the field of Optical Proximity Correction (OPC) techniques. Such models are applied so that the geometrical distance between any given point on the target and its corresponding point on the printed image is minimized. This distance is called Edge Placement Error (EPE) which has to be minimized to ensure pattern transfer fidelity [4]. Figure 1 illustrates EPE measurement on predefined locations (tap points).

Although many OPC algorithms have been proposed in the literature, such algorithms suffer from drawbacks in finding optimal solution. First of all, many previous algorithms are computationally expensive to be used for sub-100nm features due to the large number of OPC iterations needed to find solution. In realistic industrial test cases, mask data have to be produced in matter of hours to cover the huge number of design shapes needed, therefore, exhaustively searching algorithms are no more applicable to be used. Furthermore, other algorithms were developed and tested on sparse layouts while low scale dense layouts form the major part of recent lithographic yield degradation problem. Therefore, finding a best mask solution within an acceptable OPC computational time is challenging.

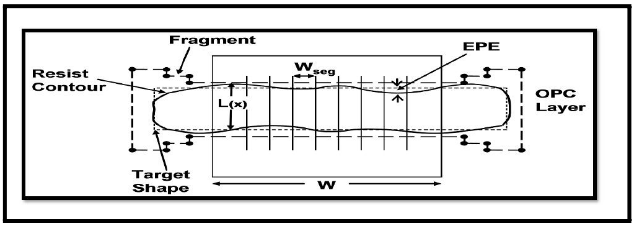


Figure 1: EPE Measurement[4]

In this paper, we propose a new mask optimization algorithm to minimize EPE within an acceptable computational time. In our algorithm, a mask pattern is optimized using two types of components: core-polygons, and serifs. The contribution of this paper is summarized as follows:

- A new intensity based algorithm is proposed to minimize EPE during OPC through iteratively modifying the core-polygons by simultaneous shifting of two neighboring segments.
- The algorithm is extended to reduce corners EPE through supporting them by serifs whose sizes iteratively modified based on approximated functions defined during preprocessing.
- Fewer number of kernels are used to generate the mask solution so that the computational time is reduced.

The rest of this paper is organized as following: In section II previous works are presented. The problem is formulated in section III. The general flow chart of our algorithm is described in section IV while system model and analyses are proposed in section V. In section VI, the entire algorithm is given. Regression experiments and algorithm experimental results are presented in section VII and section VIII concludes this paper.

II. Previous Work

In conventional OPC algorithms, a mask layout is fragmented to divide each polygon edge into freely movable segments that will be shifted outside/inside the polygon to increase/decrease the intensity in testing points till the EPE is minimized [5][6]. However, without defining the best shifting distance, the solution space will be large and thus, computationally expensive. Furthermore, shifting a segment will impact the intensity in its neighboring points. In [7], a variational OPC algorithm is proposed. In this algorithm, segment shifting is modeled as a function of variational EPE and PV-band which needs more iterations to find a best solution. A fast intensity based OPC is proposed in [8], however, this algorithm was tested only on sparse layouts without taking dense layouts into consideration.

Simultaneous Mask and Target Optimization (SMATO) was proposed in [9] with computational overhead over OPC. 1D design style to decrease mask complexity and layout decomposition for small sized features using double and triple patterning is one of the current available solutions at the cost of using more masks [10][11]. Constructing Mask Error Enhancement Factor (MEEF) matrix to guide segments shifting is one of the proposed approaches to minimize EPE. However, such approach suffers from large computational time as well. [12].

Finding the best OPC algorithm to minimize the EPE within an acceptable OPC computational time is still challenging due to the continuous feature sizes shrinking. In this paper, we try to optimize the mask layout in terms of number of EPE violations within an acceptable optimization time.

III. Problem Formulation and Layout Fragmentation

Given a target layout defined in layout region, the main objective is to find a best mask solution in layout region with the least number of EPE violations (EPEV) under nominal lithographic conditions within an acceptable optimization time.

A. Lithographic Model

In this paper, an optical model is used to generate an aerial image represented by intensity map. This model consists of optical system and projection system expressed by Transmission Cross Coefficient (TCC) which is decomposed into a set of kernels. The aerial image is obtained as shown in eq(1), where I is the aerial image, σ_k and h_k are the eigenvalue and eigenfunction set of TCC, respectively , M is the mask function, $\otimes$ denotes the convolution operation, and K is the total number of kernels. This optical model is called sum of coherent systems (SOCS) which is the preferred model for lithography [4].

$$I = \sum_{k=1}^{K} \sigma_k |(h_k \otimes M)|^2 \qquad (1)$$

The aerial image is represented by an intensity map that stores the intensity value for each pixel. Once this map is obtained, a resist model is applied to simulate photoresist exposure response. Constant Threshold Resist (CTR) is one of models used to represent the resist behavior. In this model, if the intensity is greater than intensity threshold I_{th}, the resist will be exposed followed by etching. Subsequently, the image on the wafer will be generated. In OPC, both optical and resist models are applied iteratively to generate the image contours that is compared with the target layout to decide mask adjustment for next iteration.

B. Preliminaries

Let R be a layout region which consists of pixels. Let T be a target layout in R, that is, $T \subset R$. T contains a number of polygons. A polygon $S \in T$ consists of pixels. If a pixel $p \in R$ is contained in polygon S, then we denote $p \in S$. In the following, if $p \in S \in T$, then we simply denote $p \in T$. A mask M in R which consists of pixels determines the intensity of each pixel in R under certain lithographic process conditions. The intensity of pixel $p \in R$ by mask M is denoted by $I(M, p)$. Generally, let $I(M)$ be the intensity map of R derived by M under nominal conditions. That is, $I(M)$ is a list that contains the intensity value for each pixel. Let $G(M)$ denotes the printed image of mask M on the wafer. Typically, $G(M)$ is obtained by applying a resist model. In our algorithm CTR is the model used to obtain the image. In this model, if the intensity in any pixel p is greater than or equal intensity threshold I_{th}, it will be exposed and resist will be removed. Otherwise, resist will remain. In other words, $G(M, I) = \{p \in R \mid I(M, p) \geq I_{\text{th}}\}$.

A mask generated in our algorithm consists of two types of parts; core-polygons and serifs. A core-polygon is derived from a polygon in the target layout by shifting a segment in orthogonal direction which is obtained by fragmenting a boundary edge of the polygon. A serif is a square defined at each corner of a polygon in the target layout.

C. Layout Fragmentation

The segments of a core-polygon in mask are defined by fragmenting the boundary of the corresponding polygon in the target layout according to the predefined segment length. The center of each fragmentation of the boundary is defined as a tap point as shown in Figure 2. A segment derived from fragmentation is allowed to shift in orthogonal direction of its length and is used to represent a rectangle which is added into a core-polygon in mask. The position of a segment is represented by its center where the origin is the tap point. By changing the position of a segment, the size of the corresponding rectangle added outside the target layout in mask or removed from the target layout in mask is defined.

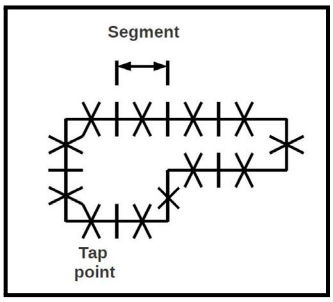


Figure 2: Layout Tap Points

D. Edge Placement Error (EPE) Formulation

EPE is the geometrical distance between any point on the target and its corresponding point on the contour of $G(M)$.

Let I_{th} denotes intensity threshold. Strictly, a mask M has no EPE if and only if $I(M,p) \geq I_{\text{th}}$, $\forall p \in T$ and $I(M,p) < I_{\text{th}}$, $\forall p \notin T$. However, such condition is infeasible to be satisfied on each pixel, therefore, a more relaxed definition is required.

Let A be the set of tap points defined in T. Let x be the allowable distance for EPE. For a tap point $t \in A$, let t^+ be a point not in T whose distance from t is x pixels which is on the line that passes t which is perpendicular to the edge of the polygon to which t belongs. Similarly, let t^- be a point in T whose distance from t is x pixels and which is on the line that passes t which is perpendicular to the edge of polygon to which t belongs. A tap point t is defined not as EPE state if $I(M,t^-) \geq I_{\text{th}}$ and $I(M,t^+) < I_{\text{th}}$. Otherwise, t is said to be as EPE state. By this way, the total number of EPE violations is given in eq(2) and eq(3).

$$epe(M,t) = \begin{cases} 1; & I(M,t^-) < I_{\text{th}} \text{ OR } I(M,t^+) \geq I_{\text{th}} \\ 0; & \text{Otherwise} \end{cases} \tag{2}$$

$$\#EPEV(M) = \sum_{t \in A} epe(M,t) \tag{3}$$

This EPE relaxed definition is used in ICCAD contest to count the number of EPE violations in terms of tap points defined on the boundary of polygons on the target layout which are similar to our algorithm [4]. In ICCAD definition, the allowable distance x for EPE is 15 pixels. However, tap points used in ICCAD contest are different from tap points used in our algorithm. Although a mask is optimized by evaluating tap points defined in our algorithm, mask is finally evaluated in terms of tap points defined in ICCAD contest to compare with other algorithms.

E. OPC Computational Time

OPC computational time is the total time required to generate the mask solution. As shown in eq(1), the mask M has to be convoluted with all kernels to obtain an aerial image. Typically, a convolution operation is computationally expensive compared to other operations. Therefore, in this paper, we model the OPC computational time as the number of convolutions (NOC) performed during optimization.

IV. OPC Algorithm Overview

Figure 3 illustrates a general flow chart of our algorithm whose input is target pattern T and output is mask M. First of all, OPC preprocessing is performed to define the main functions and parameters that drive our algorithm. In this stage, intensive data regression is applied to approximately define shifting and serif sizing functions. Thereafter, T is fragmented into segments followed by tap points definition. Then, mask generation starts with the initial mask which is same as T. Mask segments are shifted and serifs are added in the corners. This process is applied iteratively till EPE violations are eliminated or the maximum number of iterations is exceeded. Those stages are described in more details in section VI.

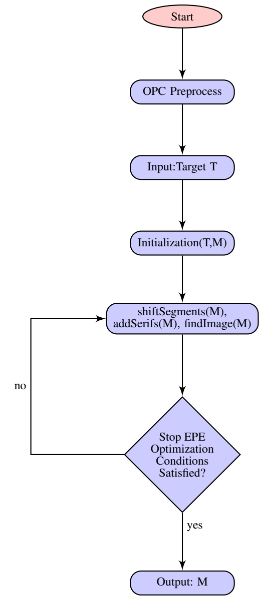


Figure 3: OPC Algorithm Flow chart

V. System Modeling and Solution

A. Segment Shifting

A simple approach is to make the intensity of a tap point the threshold value by shifting the corresponding segment. For example, shifting segment s that is corresponding to tap point t from its current position (y) to new position $(y\prime)$ would change the intensity in point t as shown in eq(4) where I_{t0} is the intensity before shifting and F_1 represents change in intensity after shifting from reference axis (which is assumed to be target layout). However, a shift of a segment affects the intensities of tap points nearby, and vice versa.

$$I_t = I_{t0} + (F_1(y\prime) - F_1(y)) \tag{4}$$

Our mask optimization algorithm consists of a number of OPC iterations in each, a core-polygon is modified by shifting neighboring segments simultaneously. In order to determine appropriate positions of two neighboring segments, the intensities of two tap points are modeled as the sum of the intensities caused by two segments and the intensity caused by the others, and these two segments are moved so that the estimated intensities of two tap points become the threshold value. The intensity caused by the corresponding segment and the intensity caused by neighboring segment are evaluated in preprocessing. The intensity caused by the others is evaluated by latest lithographic simulation when the corresponding segment and its neighbor are in their current position. Even though the accuracy is degraded after several segment positions are changed, the next lithographic simulation is not executed until all segment positions are updated to reduce the computation time. The inaccuracy is expected to become small in repetition.

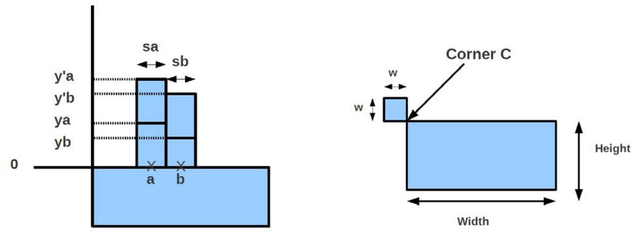
Let a and b be two neighboring tap points (as shown in Figure 4-a) defined on segments s_a and s_b, respectively. Let y_a and y_b represent the position of s_a and s_b after the k$^{\text{th}}$ OPC iteration, respectively with considering the origin on target. In the next iteration, when shifting both segments simultaneously to positions $y\prime_a$ and $y\prime_b$, the total intensity in any of the points is given in eq(5), where I_{a0} and I_{b0} represent the current intensity obtained by simulation in a and b, respectively. F_1 is used to represent the change in intensity in a point by shifting its segment while F_2 is used to represent the change in that point by shifting its neighbor. As indicated in (5), the target is to make the intensity in both points equal to I_{th}.

$$\begin{aligned} I'_a &= I_{a0} + (F_1(y\prime_a) - F_1(y_a)) + (F_2(y\prime_b) - F_2(y_b)) = I_{\text{th}} \\ I'_b &= I_{b0} + (F_1(y\prime_b) - F_1(y_b)) + (F_2(y\prime_a) - F_2(y_a)) = I_{\text{th}} \end{aligned} \tag{5}$$

By linearizing the functions $F_1(x)$ and $F_2(x)$ as in eq(6) and eq(7), we can obtain $(F_i(y\prime_a) - F_i(y_a)) = F_i(y\prime_a - y_a)) = F_i(h)$ for $i = 1,2$ where h is the distance between the segment current location and the new location after any OPC iteration. Therefore, the target is to find the solution vector (h_a, h_b) of the system shown in eq(8) with assuming (I_{a0}, I_{b0}) represents the last simulation intensities in any OPC iteration before shifting is applied and (h_a, h_b) is the shifting distance from the current position to a new position for segments s_a and s_b, respectively. To solve this system, segments shifting regression experiment was performed to define the parameters of the functions $F_1(x)$ and $F_2(x)$ (see section VII-A).

$$F_1(x) \approx L_1(x) = \alpha_1 x + \beta_1 \tag{6}$$

$$F_2(x) \approx L_2(x) = \alpha_2 x + \beta_2 \tag{7}$$



(a) Core-polygon Segments Shifting (b) Serif at Corner

Figure 4: Core-polygon and Serif

$$\begin{bmatrix} I_{a0} + F_1(h_a) + F_2(h_b) - I_{\text{th}} \\ I_{b0} + F_1(h_b) + F_2(h_a) - I_{\text{th}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{8}$$

This approach is applied to each pair of tap points in the edge. For example, let t_0, t_1, t_2, t_3 be 4 neighboring tap points in a polygon edge, the shifting distances solution vector will be found for points (t_0, t_1), then (t_1, t_2), and finally (t_2, t_3). When the solution is found for t_0 and t_1, their intensities are approximated to I_{th}, then (t_1, t_2) shifting distances is solved based on that. t_0 intensity will be affected when t_1 and t_2 are being solved, such change is detected in the next OPC iteration. Solving points as triples or more is possible. However, an accurate intensity modeling that takes shifting many segments into account needs more approximation that results in more OPC iterations.

B. Corners Serifs Sizing

A corner printability can be improved by adding a squared serif whose size is determined so that the estimated intensity of the corresponding corner becomes the threshold value. The serif insertion impact on the neighboring tap points intensities will be considered in the next OPC iteration after simulation. As shown in Figure (4-b), let c be a corner with initial simulation intensity I_{c0}, c that will be supported by serif whose width is w. When adding a serif, the intensity in c is defined as a function of the initial intensity and the serif width w as shown in eq (9) where $P(w)$ represents the changes in the corner intensity as response of serif width. Hence, serif insertion regression experiment (section VII-A) was performed to approximately linearize the function $P(w)$. By this way, we find w from eq(11) so that $I_c(I_{c0}, w) = I_{\text{th}}$. This approach can be enhanced in the future so that the segments of point a and b are shifted and serif is added simultaneously.

$$\begin{aligned} I_c(I_{c0}, w) &= I_{c0}(w) + P(w) \\ P(w) &\approx \alpha_3 w + \beta_3 \end{aligned} \tag{9}$$

C. OPC Computational Time

The intensity map is typically estimated by using a transmission cross coefficient method in which convolution operation between kernel and mask is executed, and the computation time is proportional to the number of kernels used. There is a trade-off between the accuracy of intensity map estimation and the computation time. Although a rough intensity map is obtained by using higher weight kernels in short time, the intensity of each pixel in the map tends to be smaller than the actual value and it is not accurate enough. However, with considering the relaxed EPE conditions, using large weight kernels would be sufficient for mask optimization.

Lets assume a K kernels lithographic system with N iterations required to minimize EPE, the total number of convolutions (NOC) is $K * N$. When choosing the top weight M_k kernels to be used for optimization, NOC is expected to be $M_h * N$ where $M_k < K$.

VI. Proposed Algorithm

Algorithm 1 illustrates our OPC algorithm. Using the functions and parameters obtained during preprocessing, a mask M is generated for input target T in K kernels lithographic system using two modules: First of all, initialization module, wherein, under nominal lithographic conditions, T is convoluted with the largest weight M_k kernels to calculate initial EPE violations number. Then, M is set to be same as T.

After initialization, EPE optimization module starts. In this module, if a tap point intensity is less than I_{th} or if it violates EPE relaxed conditions, system of eq(8) is applied to find the shifting distance for this point segment and its neighbor corresponding segment. After applying shifting on each tap points pair, intensity is checked in each corner and serif is added to the corner using eq(9) to find serif width. Finally, mask layout is updated, intensity map is obtained under nominal conditions using largest weight kernels, EPE violations is calculated and next iteration starts. This process will continue as long as EPE violation number is greater than 0 and the predefined maximum number of iterations is not exceeded. Abrupt increase in the number of EPE violations terminates the loop as well. Finally, the mask with minimal EPE violations is chosen as the optimal solution.

VII. Experimental Results

All experiments were executed on lithosim simulator from ICCAD contest [13]. In this simulator, T is defined in 1024×1024 pixels region where each pixel represents 1 nm $\times$ 1 nm. Nominal conditions are defined as dose 1.0 with in-focus kernels. In this simulator, $I_{\text{th}} = 0.225$. EPE is measured on the tap points defined using ICCAD formulation under nominal conditions.

A. Regression Experiments

1) Segments Shifting Experiment: Given a squared target layout (Figure 5-a) with Width=Height=400, the layout was fragmented into segments with 10 pixels length each. Then, the segment whose tap point is a started to move pixel by pixel outside the polygon (till 200 pixels) and inside the

TABLE I: EPE and Computational Time for Public Benchmarks

Benchmark	#EPEV (before)	#EPEV (after)	Time (sec)	NOC
M1-test1	158	4	230	35
M1-test2	167	4	217	35
M1-test3	132	34	237	35
M1-test4	122	0	204	30
M1-test5	252	0	201	30
M1-test6	195	0	201	30
M1-test7	293	0	203	30
M1-test8	107	0	151	20
M1-test9	243	0	207	30
M1-test10	72	0	142	20

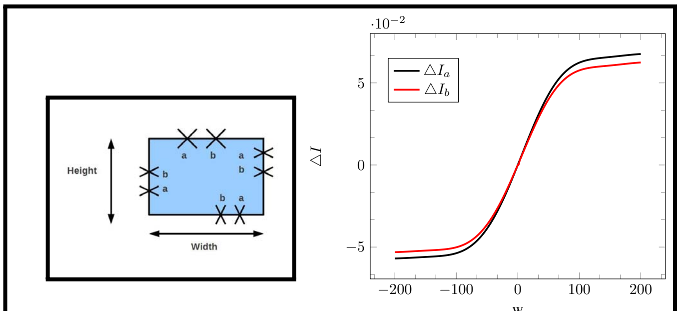
Algorithm1: OPC Algorithm

```
*********************** Initialization**************************
M_k ← chooseMKernelsFromAll(K) // largest weight kernels
I_Mk(T) ← findIntensityMapUsingMKernel(T, nominalDose, inFocus)
M ← applyFragmentation(T)
M ← T // Initial Mask is target pattern
I(M) ← I_Mk(T)
S ← getSegmentsFromMask(M)
T ← findSegmentCenters(S)
C ← findCornerns(M)
Shift[S] ← 0 // Initially, shift distances are zeros
Serif[C] ← 0 // Initially, serif widths are zeros.
G(M) ← applyResistModel(I(M), I_th)
epe ← calculateEPE(G(M), T)
*********************** EPE Minimization Module**************************
while epe > 0 and iteration < MAX_NUM_Of_ITERATIONS and
(epe[iteration] − epe[iteration − 1]) < k do
    for each edge E in M do
        for each tap point t_i in E do
            if I(M, t_i) < I_tht OR I(M, t^−) < I_th OR I(M, t^+) ≥ I_th then
                SolveSystem(I(M, t_i), I(M, t_{i+1}), h_0, h_1, I_th)
                Shift[S_i] ← Shift[S_i] + h_0
                Shift[S_{i+1}] ← Shift[S_{i+1}] + h_1
                I(M, t_i) ← I_th
                I(M, t_{i+1}) ← I_th
            end if
        end for
    end for
    for each corner c_i do
        if I(M, c_i) < I_th then
            SolveSystem(I(M, c_i), w, I_th)
            serif[c_i] ← serif[c_i] + w
        end if
    end for
    M[iteration] ← updateMaskShape(M, shift, serif)
    I(M) ← findIntensityMapUsingMKernel(T, nominalDose, inFocus)
    G(M) ← applyResistModel(I(M), I_th)
    epe ← calculateEPE(G(M), T)
    iteration ← iteration + 1
end while
M ← M[iteration] // choose mask with the least EPE violations
```

polygon (till -200 pixels). The intensity in a and b was recorded before shifting (I_{a0}, I_{b0}). Then, after each segment movement, the change in intensity $\triangle I_a = I_a - I_{a0}, \triangle I_b = I_b - I_{b0}$ was recorded. For more data reliability, this experiment was performed on different a and b locations as shown in Figure 5-a and the total results were averaged. Figure 5-b shows ($\triangle I_a, \triangle I_b$) versus the shifting distance (h). Note that the black curve represents the changes in the tap point intensity while the red curve shows the changes in the neighboring point intensity.

Hence, the functions $F_1(h)$ and $F_2(h)$ (shown in eq(5)) are approximated following the black and red curves respectively (Figure 5-b). However, our experiments show that in most cases, the shifting distance is expected to be in the transient region of the curve, i.e, $h \in [-100, 100]$. Therefore, the functions $F_1(h)$ and $F_2(h)$ are linearized as shown in eq(6) and eq(7) where $\alpha_1 \approx 0.00913$, $\alpha_2 \approx 0.00835$, $\beta_1 \approx 0.00004$ and $\beta_2 \approx 0.00029$ according to our regression results.

2) Serif Insertion Experiment: In this experiment, a rectangular mask shape (as shown in Figure 4-b) layout with Width=400 and Height=100 was supported by squared serif



(a) Experiment Setup (b) Shifting Distance VS $\triangle I$

Figure 5: Segments Shifting Experiment

of width w. From $w = 10$ pixels, the serif width started to increase pixel by pixel and in each time, the intensity change ($\Delta I_c = I_c - I_{c0}$) in the corner was recorder. Figure 6 shows the plot of data obtained from which we linearize the function $P(w)$ as in eq(9) with $\alpha_3 \approx 0.00187$, $\beta_3 \approx -0.01337$.

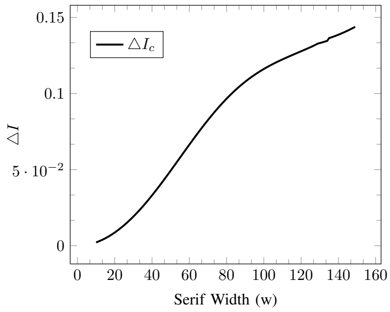


Figure 6: Serif Width VS $\triangle I$

B. Algorithm Experimental Results

The proposed algorithm was implemented in C on a 4 cores 3.6 GHz Linux machine with total memory of 1986912 KB. The algorithm was built on top of lithosim and executed on the 10 public benchmarks released by IBM for ICCAD contest 2013. In all experiments, 10 pixels segment length was chosen to be reasonable for dense and sparse benchmarks, and the maximum number of iterations is 8. However, more iterations might be needed for more complex layouts as long as no abrupt increase in EPE occurs. For each benchmark, the number of EPE violations (#EPE) was measured in addition to the execution time in seconds. In table 1, the number of EPE violations before optimization (initial image when using target layout as the mask), after optimization, optimization time in seconds, and the number of convolutions are shown for each benchmark. The total number of EPE violations was measured using EPE checker tool from ICCAD contest [13]. In lithosim, 24 kernels are required to generate the image while we use only first 5 kernels in our optimization algorithm. It is obvious that EPE was eliminated in most of the benchmarks within an acceptable computational time. Figure 7 illustrates benchmark 9 target layout (Figure 7-a), image before optimization (Figure 7-b), generated mask using our algorithm (Figure 7-c), and image after optimization in which no EPE violations occur. (Figure 7-d). Figures 8 and 9 illustrate benchmark 1 and benchmark 9 EPE convergence with different iterations, respectively.

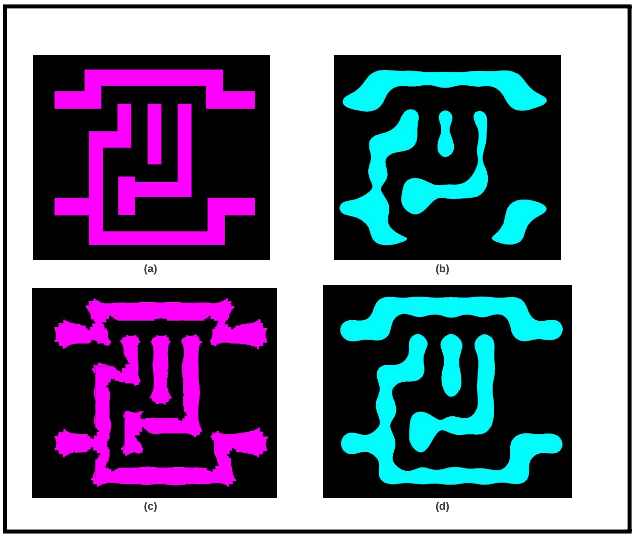


Figure 7: M1-test9: (a)Target. (b)Initial Image. (c)OPC Mask. (d)Final Image

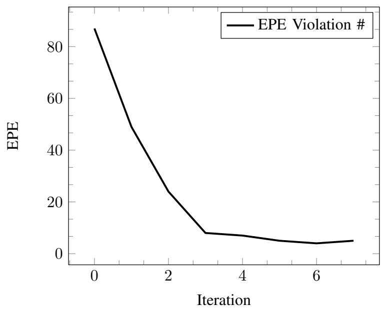


Figure 8: M1-test1 EPE Convergence

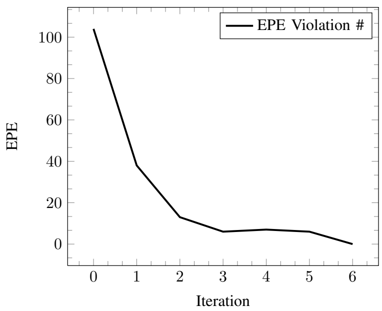


Figure 9: M1-test9 EPE Convergence

VIII. Conclusion and Future Work

In this paper, we proposed a new algorithm to minimize EPE within an acceptable computational time. Shifting and serif insertion based on systems solutions are used to minimize EPE. Only the largest weight kernels are used to obtain the intensity map during each OPC iteration. Our simulations show that our algorithm eliminated in EPE in most of the public benchmarks used in ICCAD contest within a reasonable computational time. In the future, this algorithm will be extended to consider process variability, and mask manufacturability so that less complex mask shapes can be obtained.

Acknowledgement

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B)25280013.

References

[1] Xu Ma and Gonzalu Arce, *"Computational Lithography"*, Wiley Publisher, 2010.

[2] R.Fabian Pease and Stephen Chou, *"Lithography and Other Patterning Techniques for Future Electronics"*, proceedings of the IEEE, 2008, pp.248-270.

[3] Allan Gu and Avideh Zakhor, *"Optical Proximity Correction With Linear Regression"*, IEEE Transactions on Semiconductor Manufacturing, 2008, pp.263-271.

[4] Shayak Banerjee, Zhou Li and San Nassif, *"ICCAD-2013 CAD Contest in Mask Optimization abd Benchmark Suite"*, proceedings of ICCAD, 2013, pp.271-274.

[5] Ayman Hamouda, Mohab Anis, and Karim Karim, *"Using Segmented Models for Initial Mask Perturbation and OPC Speedup"*, Proc. SPIE 8880, Photomask Technology, 2013.

[6] Tim Fuhner and Andreas Erdmann, *"Improved Mask and Source Representations for Automatic Optimization of Lithographic Process Conditions Using a Genetic Algorithm"*, proceedings of SPIE 5754, Optical Microlithography, 2005.

[7] Peng Yu, Sean Shi, and David Pan, *"Process Variation Aware Optical Proximity Correction with Variational Lithography Modeling"*, proceedings of DAC, 2006, pp.785-790.

[8] Peng Yu and David Pan, *"A Novel Intensity Based Optical Proximity Correction Algorithm with Speedup in Lithography Simulation"*, proceedings of ICCAD, 2007, pp.854-858.

[9] Shayak Banerjee, Kanak Agarwal, and Michael Orshansky, "SMATO: Simultaneous Mask and Target Optimization for Improving Lithographic Process Window", proceedings of ICCAD, 2010, pp.100-106.

[10] Michael Smayling, "1D Design Style Implications for Mask Masking and CEBL", Proc. SPIE 8880, Photomask Technology, 2013.

[11] Andrew Kahng, Chul-Hong Park, Xu Xu, and Hailong Yao, "Layout Decomposition Approaches for Double Patterning Lithography", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems", 2010, pp.939-952

[12] Nick Cobb and Yuri Granik, and David Pan, *"Model-based OPC using the MEEF matrix"*, proceedings of 22nd Annual BACUS Symposium on Photomask Technology, 2002, pp.1281-1292.

[13] http://cad_contest.cs.nctu.edu.tw/CAD-contest-at-ICCAD2013/problem_c/