論文 / 著書情報
Article / Book Information

Title	A Fast Mask Manufacturability and Process Variation Aware OPC Algorithm with Exploiting a Novel Intensity Estimation Model
Authors	Ahmed Awad, Atsushi Takahashi, Chikaaki Kodama
出典 / Citation	IEICE Trans. Fundamentals, Vol. E99-A, No. 12, pp. 2363-2374
発行日 / Pub. date	2016, 12
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2016 Institute of Electronics, Information and Communication Engineers.

PAPER *Special Section on VLSI Design and CAD Algorithms*

A Fast Mask Manufacturability and Process Variation Aware OPC Algorithm with Exploiting a Novel Intensity Estimation Model

Ahmed AWAD[†a)], ***Member***, **Atsushi TAKAHASHI**[†], ***and*** **Chikaaki KODAMA**[††], ***Senior Members***

SUMMARY With being pushed into sub-16 nm regime, advanced technology nodes printing in optical micro-lithography relies heavily on aggressive Optical Proximity Correction (OPC) in the foreseeable future. Although acceptable pattern fidelity is utilized under process variations, mask design time and mask manufacturability form crucial parameters whose tackling in the OPC recipe is highly demanded by the industry. In this paper, we propose an intensity based OPC algorithm to find a highly manufacturable mask solution for a target pattern with acceptable pattern fidelity under process variations within a short computation time. This is achieved through utilizing a fast intensity estimation model in which intensity is numerically correlated with local mask density and kernel type to estimate the intensity in a short time and with acceptable estimation accuracy. This estimated intensity is used to guide feature shifting, alignment, and concatenation following linearly interpolated variational intensity error model to achieve high mask manufacturability with preserving acceptable pattern fidelity under process variations. Experimental results show the effectiveness of our proposed algorithm on the public benchmarks.

key words: *pattern fidelity, process variation, mask manufacturability, mask data volume, computation time*

1. Introduction

With the continuous shrinkage of advanced technology nodes into sub-16 nm features, proximity effects become more pronounced in a lithographic process, wherein, an Integrated Circuit (IC) nodes are patterned layer by layer on the silicon wafer. As a result, wafer image suffers from distortions which cause pattern fidelity degradation as illustrated in Fig. 1. Moreover, a raw lithographic process is susceptible to process variations which impact chip timing/power and thus cause overall yield degradation [1], [2].

To improve pattern fidelity under process variations, the industry relies heavily on Optical Proximity Correction (OPC), wherein, a mask pattern is iteratively adjusted followed by wafer image computation for mask evaluation till sufficient pattern fidelity is achieved. Next Generation Lithography (NGL) is still not in the track, therefore, OPC is still the main stream in printing sub-16 nm features [3], [4].

However, high level of aggressiveness in OPC algorithms is required to satisfy sufficient pattern fidelity for advanced technology nodes. This results in large OPC computation time which makes an OPC algorithm infeasible for realistic industrial cases [5]. Furthermore, mask manufacturability degradation is expected due to the complex outputted mask solutions such as Inverse Lithography Technology (ILT) masks as shown in Fig. 2 [6], [7].

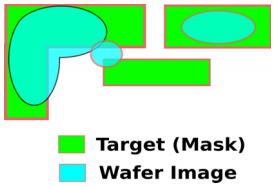


Fig. 1 Wafer image distortions.

Fig. 2 Complex ILT mask [6].

Wafer image computation during OPC is the principal contributor for OPC computation time. Thus, several wafer estimation models have been proposed. Intensity Difference Map (IDM) has been introduced to estimate the intensity map (from which wafer image is extracted) of a given mask within a short a time and with acceptable estimation accuracy [8]. IDM stores the intensity map of some reference mask estimated by lower weight Sum of Coherent Systems (SOCS) kernels [9] of a lithographic system. IDM iteratively compensates the intensity map of a mask estimated using only top weight kernel. However, this usage of IDM assumes that it is invariant of mask shape slight changes whereas actual intensity value for lower weight kernels is kernel and local mask structure dependent.

Design aware OPC algorithms include a set of mask design rules in the OPC recipe for higher output mask manufacturability. Design rules define a set of minimal allowable dimensions in the mask geometry, such as minimum edge length [10]. However, satisfying design rules during OPC causes intensity estimation error whose neglecting turns out into pattern fidelity degradation.

In this paper, we propose an OPC algorithm to find a highly manufacturable mask solution for a target pattern within a short computation time with preserving acceptable pattern fidelity under process variations. Our contributions are summarized as follows:

- An intensity estimation model is proposed to speedup wafer image computation time. In this model, inten-

Manuscript received March 10, 2016.
Manuscript revised July 1, 2016.
[†]The authors are with Tokyo Institute of Technology, Tokyo, 152-8550 Japan.
[††]The author is with Toshiba Corporation, Yokohama-shi, 247-8585 Japan.
a) E-mail: awad.ahmed@eda.ce.titec.ac.jp
DOI: 10.1587/transfun.E99.A.2363

sity relation with local mask density and kernel type is exploited to adaptively improve the accuracy of IDM. Thus, IDM is adapted following mask shape changes which turns out into more accurate intensity estimation.
- An intensity error modeling is proposed to iteratively interpolate the deviation of intensity of a given point from the target intensity as a function of feature shifting distance to guide next adjustment under the assumption of error invariance of mask shape slight changes.
- An intensity based OPC algorithm driven by our intensity estimation model is proposed. In this algorithm, mask is iteratively adjusted following intensity error in such a way that high mask manufacturability is achieved through minimizing mask design rule violations with preserving acceptable pattern fidelity under process variations due to intensity error compensation.
- A post OPC phase is proposed to improve mask manufacturability and reduce mask data volume through applying a set of rules that control mask adjustment to preserve acceptable pattern fidelity.

The rest of this paper is organized as follows: Previous work is presented in Sect. 2. Preliminaries and mask evaluation parameters are proposed in Sect. 3. Intensity estimation model and intensity error modeling are explained in Sect. 4 and Sect. 5, respectively. Our proposed OPC algorithm and its modules are proposed in Sect. 6. Experimental results are presented in Sect. 7 while Sect. 8 concludes this paper.

2. Previous Work

Several algorithms have been proposed to improve mask manufacturability in the field of design aware OPC [11]. Pre-OPC hotspots detection and fixing has been proposed in [12] to reduce mask complexity at the cost of complexity increase with iterations. Other algorithms include a set of design rules in the OPC as published in [10] at the cost of computation time increase. PVOPC algorithm proposes a post OPC stage to satisfy mask notch rule [13]. Eliminating jogs during Sub-Resolution Assisting Features (SRAFs) insertion has been proposed through smoothing the target pattern at the cost of computation time increase [14], [15].

To accelerate OPC computation time, several intensity estimations have been proposed following Sum of Coherent Systems (SOCS) model [9]. Linearized model to speedup simulation time has been proposed in [16] at the cost of accuracy loss for dense patterns. Simulating less critical regions using lower number of kernels has been proposed in [17] at the cost of computation time increase with iterations. Intensity Difference Map (IDM) has been proposed [8], wherein, top weight kernel is used to simulate intensity map for which compensative IDM is added. However, IDM itself needs to be adaptively fixed following mask shape chances.

The work presented in this paper extends the algorithm proposed in [8] in several aspects: First of all, adaptive intensity difference map is introduced for better intensity estimation. Second, mask manufacturability and mask data volume are numerically evaluated and improved in the proposed algorithm. Finally, variational intensity error modeling is included in the OPC recipe to keep algorithm convergence controllable in terms of pattern fidelity under process variations with satisfying mask design rules. A primary version of this work has been published in [18]. However, our work in [18] does not consider process variations and feature alignment to eliminate mask notch rule violations.

3. Preliminaries and Mask Evaluation Metrics

3.1 Problem Description

Given a target pattern defined in a region of pixels, the objective is to find a mask solution with high manufacturability within a short computation time and with high pattern fidelity and robustness against process variations.

3.2 Lithographic Terminology

Let R be a layout region of pixels. Let T and M be a target pattern and a mask pattern, respectively, defined in region R such that $T \subset R$ and $M \subset R$. Both T and M consist of a set of non-overlapped rectilinear polygons where a polygon S consists of a number of connected pixels. If a pixel $p \in S \in T$, we simply denote it by $p \in T$. Similarly, if $p \in S \in M$, we denote it by $p \in M$.

The boundary of a polygon S consists of a number of edges where an edge is a horizontal or vertical line connecting two corners in the boundary of S. Let E_{T} and E_{M} denote the set of edges defined along the boundary of the polygons included in T and M, respectively. Let $l(e)$ be the length of edge e and $D(e_i, e_j)$ be the Manhattan distance between edges e_i and e_j.

In this paper, the polygons of a mask M are classified into three types: Core-polygons, serifs, and Sub-Resolution Assisting Features (SRAFs). A core-polygon is constructed by adding/removing rectangles on/from parts of the boundary of its corresponding polygon in T. A serif is a squared feature added onto a corner in T. An SRAF is a long bar inserted parallel to the edges of the polygons in T. Figure 3(a) shows a target pattern and Fig. 3(b) illustrates an OPCed mask of that target and its components.

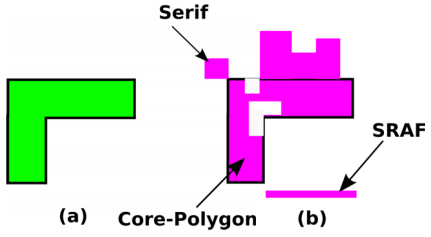


Fig. 3 (a) Target T. (b) Mask M.

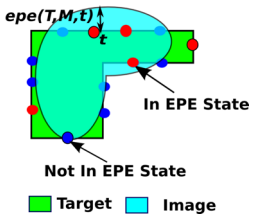


Fig. 4 Pattern fidelity evaluation.

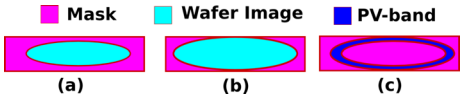


Fig. 5 (a) Innermost image. (b) Outermost image. (c) PV-band area.

3.3 Lithographic Model

Sum of Coherent Systems (SOCS) model is used in this paper to represent the intensity map through a mask in region R. In this model, optical system is decomposed into a set of kernels. Each kernel has an eigenfunction which represents its filtering behavior and eigenvalue which represents its weight for intensity estimation. For a mask M, intensity map $I(M, K)$ is defined as given in Eq. (1) where K denotes the set of all kernels in a lithographic system, σ_k and ϕ_k represent the eigenvalue and eigenfunction of kernel $k \in K$, respectively, and $\otimes$ denotes convolution operation [9].

$$I(M, K) = \sum_{k \in K} \sigma_k |(\phi_k \otimes M)|^2 \tag{1}$$

From intensity map, wafer image is extracted by applying Constant Threshold Resist (CTR) model. In this model, intensity threshold of exposure I_{th} is predefined. All pixels whose intensity value is greater than or equal to I_{th} form the wafer image $G(M, K)$ as given in Eq. (2) where $I(M, K, p)$ represents intensity value in pixel $p \in R$ through mask M.

$$G(M, K) = \{p \in R \mid I(M, K, p) \geq I_{\text{th}}\} \tag{2}$$

In this paper, K denotes the set of all kernels in a lithographic system. Thus, when intensity map is estimated using all kernels, K is omitted such that $I(M)$ is equivalent to $I(M, K)$ and $I(M, p)$ represents intensity value in pixel p. Otherwise, the set of kernels by which mask M is convoluted is explicitly stated such as $I(M, K')$ where $K' \subset K$ and $I(M, K', p)$ represents the intensity in pixel p using set of kernels K'. Same notation is applied for wafer image.

3.4 Mask Evaluation Metrics

Let P_w be a process window that contains a set of process conditions upon request. The most likely process condition within P_w is defined as nominal condition under which nominal intensity map $I_n(M)$ is defined. Innermost intensity map $I_i(M)$ is the set of minimum intensity values within P_w while outermost intensity map $I_o(M)$ is the set of maximum intensity values. Nominal image $G_n(M)$, innermost image $G_i(M)$, and outermost image $G_o(M)$ are extracted from $I_n(M)$, $I_i(M)$, and $I_o(M)$, respectively.

3.4.1 Mask Pattern Fidelity

Pattern fidelity of a mask is evaluated in terms of Edge Placement Error (EPE) under nominal process condition. EPE is the geometrical distance between a point on the boundary of T and its corresponding point onto the wafer image. Let $epe(M, t)$ denote the EPE in point $t \in T$ and let $epe_{\max}$ denote the maximum allowable EPE.

Let A denote a set of evaluation points defined in the boundary of T. $\forall t \in A$, if $epe(M, t) > epe_{\max}$, t is said to be in EPE state, otherwise, it is not. The total number of EPE valuations of a mask M, denoted by $\#EPEV(M)$, is the number of evaluation points in EPE state. Pattern fidelity of M is assumed to be inversely proportional to $\#EPEV(M)$. Figure 4 illustrates pattern fidelity evaluation.

3.4.2 Mask Robustness against Process Variations

Mask robustness against process variations is evaluated in terms of Process Variability band (PV-band) area. PV-band area of a mask M, denoted by $PV(M)$, is defined by XORing innermost and outermost wafer images in P_w as given in Eq. (3). Figure 5 illustrates PV-band area for a given mask. Larger PV-band area indicates a less mask robustness robustness against lithographic process variations.

$$PV(M) = |\{p \in R \mid G_o(M, p) \oplus G_i(M, p)\}| \tag{3}$$

3.4.3 Mask Manufacturability

Satisfying mask design rules guarantees higher mask manufacturability. In this paper, mask notch and spacing rules are considered such that mask manufacturability is assumed to be inversely proportional to the violations number of those rules. Figure 6(a) shows some examples of such violations.

Mask notch rule defines the minimum allowable edge (notch) length in the boundary of the polygons in the mask, denoted by d_{notch}. Any edge whose length is less than d_{notch} violates this rule. The number of notch rule violations of mask M, denoted by $\#NotchV(M)$, is given in Eq. (4).

$$\#NotchV(M) = |\{e \mid e \in E_{\text{M}}, l(e) < d_{\text{notch}}\}|. \tag{4}$$

Two edges in the boundary of a mask are said to be a comparison pair if they belong to two different polygons or if they belong to the same polygon without overlapping between them as illustrated in Fig. 6(b). Spacing rule defines the minimum allowable distance between edges within a comparison pair, denoted by $d_{\min}$. Let edges $e_i \in E_{\text{M}}$ and $e_j \in E_{\text{M}}$ form comparison pair (e_i, e_j), spacing rule says that $D(e_i, e_j) < d_{\min}$. The number of spacing rule violations, denoted by $\#SpaceV(M)$, is given in Eq. (5) where C_p represents the set of comparison pairs in M. Comparison pairs of a mask can be retrieved by bounding techniques.

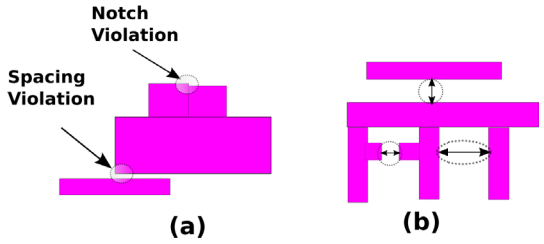


Fig. 6 (a) Design rule violations. (b) Comparison pair examples where the arrows represent the distance between each two edges in a pair.

$$\#SpaceV(M) = |\{(e_i, e_j) \mid (e_i, e_j) \in C_{\mathrm{p}}, D(e_i, e_j) < d_{\min}\}| \tag{5}$$

4. Intensity Estimation Model

There is a trade-off between computation time and intensity estimation accuracy. The more kernels by which a mask is convoluted during estimation, the more estimation accuracy. However, computation time increases due to the computationally expensive convolution operations.

On the other hand, using fewer number of kernels would degrade intensity estimation accuracy specially around mask boundaries. Typically, Fast Fourier Transform (FFT) is applied to convert the mask function from spatial domain into frequency domain. The edges of a polygon are known as the set of pixels with highest frequencies if compared with other pixels nearby the polygon center. Thus, during OPC, when wafer image is estimated with SOCS model, lower weight kernels work as low pass filters. Such filters would suppress those high frequencies around the target boundaries. Using fewer number of kernels would ignore the impact of such filters and thus, results in insufficient intensity estimation. Such estimation will improperly guide the OPC response during optimization.

Thus, our intensity estimation model aims to minimize the number of convolutions in intensity estimation with preserving acceptable estimation accuracy. Acceptable estimation means an estimation of the intensity which can properly guide the OPC response to generate a good mask solution in terms of pattern fidelity under process variations.

4.1 Key Idea of the Proposed Intensity Estimation Model

Given a set of kernels, the intensity of a pixel against local mask density around that pixel has been examined, and a linearity is observed as shown in Fig. 7. In our proposed intensity estimation, this linearity is exploited to estimate the intensity through a mask in a certain pixel using a reference mask. To accelerate this estimation, layout region is divided into grids for each average intensity is estimated. The linearity coefficient is grid and kernel dependent. For example, this coefficient for one grid can be positive for some kernels while negative for others. Similarly, for one kernel, this coefficient can be positive for some grids and negative for others.

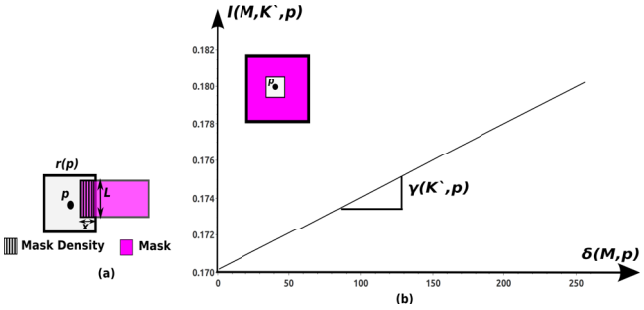


Fig. 7 (a) Local mask density. (b) Relation between intensity and mask density in pixel p shown when filling the local region of this pixel (16×16 nm region) by mask pixels (Simulations were executed using Lithosim simulator [19]).

Therefore, lower weight kernels are grouped following region dependency property which turns out into reducing the number of linearity coefficients to be one per grid and another one common for all grids in the layout region.

4.2 Previous Intensity Estimation and Main Drawbacks

Intensity Difference Map (IDM) is the difference between two intensity maps obtained using two sets of SOCS kernels [8]. Let K' be a set of kernels such that $K' \subset K$, IDM of mask M, denoted by $I_{\mathrm{diff}}(M, K')$, is given in Eq. (6). $I_{\mathrm{diff}}(M, K', p)$ denotes the IDM value in pixel p.

$$I_{\mathrm{diff}}(M, K') = I(M) - I(M, K') \tag{6}$$

Intensity map of a mask M is estimated using a reference mask M_{ref} with exploiting IDM. The assumption is that both M and M_{ref} are masks for the same target pattern T. Let $k_0 \in K$ be the top weight kernel. Intensity map of mask M is estimated through compensating $I(M, \{k_0\})$ with the IDM of M_{ref} as given in Eq. (7).

$$I(M) \approx I(M, \{k_0\}) + I_{\mathrm{diff}}(M_{\mathrm{ref}}, \{k_0\}) \tag{7}$$

In our work published in [8], T is used as M_{ref} to estimate the intensity map of mask M during OPC. However, since high frequencies around the boundaries of M_{ref} (boundaries of T in this case) are suppressed by the low weight kernels, the information that $I_{\mathrm{diff}}(T, \{k_0\})$ stores around target boundaries suffers from lack of accuracy which degrades overall estimation accuracy. This issue can be tackled through applying one OPC step on T to generate an initial OPCed mask solution M_0. With using M_0 as M_{ref}, the frequencies around the target boundaries are reduced which improves the information that $I_{\mathrm{diff}}(M_0, \{k_0\})$ stores around those boundaries. Therefore, $I_{\mathrm{diff}}(M_0, \{k_0\})$ works as a compensative map during OPC.

This intensity estimation uses IDM as a static compensative map with assuming its invariance property against mask shape slight changes. However, during OPC, mask shape iteratively deviates from M_{ref} and thus estimation accuracy gradually decays. Therefore, IDM needs to be adaptively corrected following mask shape changes.

4.3 Intensity Relation with Local Mask Density

Let $p \in R$ be a pixel centering region $r(p) \subset R$ as shown in Fig. 7(a). The local density of mask M in pixel p, denoted by $\delta(M, p)$, is the number of mask pixels intersecting region $r(p)$, i.e. $\delta(M, p) = |M \cap r(p)|$. The relation between intensity of pixel p estimated using set of kernels $K' \subset K$ and $\delta(M, p)$ is shown in Fig. 7(b). The linearity constant of this relation, denoted by $\gamma(K', p)$, is called the correction coefficient which is pixel region and kernels set dependent.

4.4 Intensity Estimation with Adaptive IDM

With exploiting the relation between intensity and local mask density, the intensity of a pixel p through mask M is estimated using mask M_{ref} by compensating top weight kernel intensity map by adding $I_{\text{diff}}(M_{\text{ref}}, \{k_0\}, p)$ which is adapted by considering the mask density difference between masks M and M_{ref} and the correction coefficient of p as given in Eq. (8) where $k_0 \in K$ is the top weight kernel.

$$\begin{aligned} I(M, p) \approx & I(M, \{k_0\}, p) + I_{\text{diff}}(M_{\text{ref}}, \{k_0\}, p) \\ & + \gamma(K \setminus \{k_0\}, p) * (\delta(M, p) - \delta(M_{\text{ref}}, p)) \end{aligned} \tag{8}$$

4.5 Coefficient Interpolation and Kernels Classification

Let M_i and M_j be two masks of target pattern T, the correction coefficient of a pixel p using arbitrary set of kernels K' is linearly interpolated using both masks as given in Eq. (9).

$$\gamma(K', p) = \frac{I(M_j, K', p) - I(M_i, K', p)}{\delta(M_j, p) - \delta(M_i, p)} \tag{9}$$

Roughly saying, if higher weight kernels contribute in intensity estimation, correction coefficient of a pixel is expected to be local region location dependent. This set of kernels is called Region Variant Kernels (RVKs). If only lower weight kernels contribute in intensity estimation, correction coefficient is expected to be region location independent. This set of lower weight kernels is called Region Invariant Kernels (RIKs).

SOCS Kernels classification can be utilized through predefining a threshold for the eigenvalue (weight). Let σ_{th} denote this threshold where σ_{th} is a fraction of the top weight kernel eigenvalue. A kernel $k_i \in K$ is classified as RVK if $\sigma_{k_i} \geq \sigma_{\text{th}}$, otherwise, it is classified as RIK. Top weight kernel is excluded from this classification. As a case of study, in Lithosim simulator [19], the set of kernels include $K = \{k_0, k_1, k_2, \ldots, k_{23}\}$. The corresponding eigenvalues for those kernels are $\{84, 35, 34, 14, \ldots, 0.44\}$. As an example, with setting σ_{th} to be one-third of the top weight kernel eigenvalue, $\{k_1, k_2\}$ will be the set of RVKs while $\{k_3, k_4, \ldots, k_{23}\}$ will be the set of RIKs.

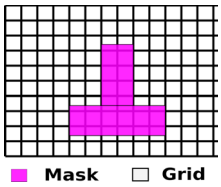


Fig. 8 Layout region gridding.

Let $k_0 \in K$ be the top weight kernel. The total set of kernels is formulated as $K = \{k_0\} \cup K_{\text{rv}} \cup K_{\text{ri}}$ where K_{rv} and K_{ri} represent the set of RVKs and RIKs, respectively. In that sense, two types of correction coefficients of a pixel p are utilized: region invariant coefficient, denoted by $\gamma_{\text{ri}}(K_{\text{ri}})$, and region variant coefficient, denoted by $\gamma_{\text{rv}}(K_{\text{rv}}, p)$. γ_{ri} is assumed to be the same for all pixels and obtained during preprocessing while $\gamma_{\text{rv}}(K_{\text{rv}}, p)$ is linearly interpolated per pixel using two masks as in Eq. (9) with setting $K' = K_{\text{rv}}$.

4.6 Overall Intensity Estimation Modeling

Interpolating a correction coefficient per pixel in layout region is computationally expensive. Therefore, the layout region R is divided into squared grids, for each, average intensity and mask density are considered in calculating per grid correction coefficient. Fig. 8 illustrates gridding process.

Let $W = \{w_0, w_1, \ldots, w_n\}$ denote the set of grids such that the area of each grid is $Q \times Q$. Grid size should not be large since mask geometry impact within the grid becomes significant on intensity estimation accuracy. Let $\delta(M, w) = |M \cap w|$ denote the mask density of and $\widehat{I}(M, w)$ denote the average intensity of grid w as given in Eq. (10).

$$\widehat{I}(M, w) = \sum_{p \in w} I(M, p)/|w| \tag{10}$$

A correction coefficient of grid w using set of kernels K', denoted by $\gamma(K', w)$, is linearly interpolated using two masks like M_i and M_j as given in Eq. (11).

$$\gamma(K', w) = \frac{\widehat{I}(M_j, K', w) - \widehat{I}(M_i, K', w)}{\delta(M_j, w) - \delta(M_i, w)} \tag{11}$$

With exploiting layout region gridding and kernels classification, the intensity in each pixel is estimated using our model as given in Eq. (12) where w_i is the grid for which pixel p belongs, $K = \{k_0\} \cup K_{\text{rv}} \cup K_{\text{ri}}$, and $\triangle\delta(M, M_{\text{ref}}, w_i) = \delta(M, w_i) - \delta(M_{\text{ref}}, w_i)$.

$$\begin{aligned} I(M, p)\,|_{p \in w_i} \approx & I(M, \{k_0\}, p) + I_{\text{diff}}(M_{\text{ref}}, \{k_0\}, p) \\ & + \gamma(K_{\text{ri}}) * \triangle\delta(M, M_{\text{ref}}, w_i) \\ & + \gamma(K_{\text{rv}}, w_i) * \triangle\delta(M, M_{\text{ref}}, w_i) \end{aligned} \tag{12}$$

5. Intensity Error Modeling for Mask Adjustment

Mask adjustment following intensity based OPC algorithms aims to make the intensity for each point along the target

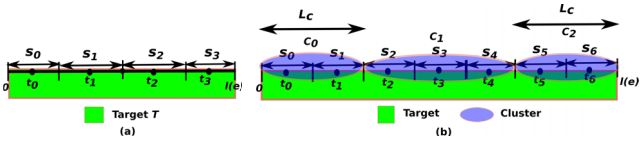


Fig. 9 (a) Fragmenting an edge into segments. (b) Grouping segments into clusters.

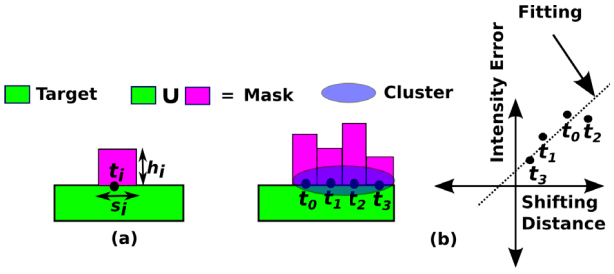


Fig. 10 (a) Shifting a segment. (b) Intensity error fitting.

boundary equivalent to I_{th} [20]. However, the intensity of a point might be deviated from I_{th} due to the impact of other features nearby and the difficulty in finding exact integer solution for adjusting distances. With assuming intensity deviation amount from I_{th} for a point to be the same if a mask pattern is slightly modified, estimating this amount for a given point would help in guiding next adjustments of the mask such that the expected deviation amount is minimized.

Each edge along the boundary of T is fragmented into movable segments. Segment length, denoted by L_{seg}, is predefined such that $L_{\text{seg}} \geq d_{\text{notch}}$ to satisfy mask notch rule. Shifting a segment is corresponding to adding/removing a rectangle to/from polygons in T. The center of each segment in the boundary of T is defined as a tap point. For a segment s_i, let t_i denote the tap point of this segment as shown in Fig. 9(a).

Intensity error of a tap point t_i, denoted by $E(M, t_i, h_i)$, is defined as the deviation of the intensity of t_i from I_{th} as given in Eq. (13), where $I(M, t_i, h_i)$ denotes the intensity of t_i after shifting its segment by h_i pixels from its location in the boundary of T as shown in Fig. 10(a).

$$E(M, t_i, h_i) = I(M, t_i, h_i) - I_{\text{th}} \tag{13}$$

Intensity error behavior is segment location dependent within the edge. Furthermore, it is affected by the polygons surrounding the segment. To estimate intensity error as a function of segment shifting distance, the segments belonging to a target polygon edge are grouped into clusters as shown in Fig. 9(b). For all segments belonging to a cluster, the best linear relation fitting the shifting distances versus intensity errors for those tap points is interpolated as illustrated in Fig. 10-b. Each segment within a cluster follows the estimated intensity error as a function of its shifting distance in next mask adjustment iteration as will be explained.

6. Proposed OPC Algorithm

Our proposed OPC algorithm flowchart is shown in Fig. 11.

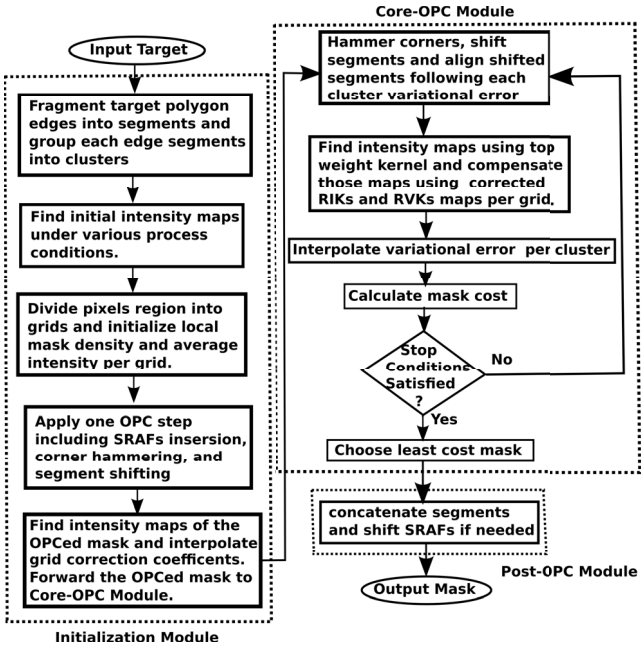


Fig. 11 OPC algorithm flowchart.

The input is target T and output is mask M. This algorithm consists of three modules: Initialization, Core-OPC, and Post-OPC. Prior to OPC algorithm, a preprocessing is applied in which kernels are classified into RIKs and RVKs.

6.1 Initialization Module

The purpose of this module is to find an initial mask solution for target T. This initial solution is exploited to interpolate the correction coefficient for each grid to be included in intensity estimation during the iterative Core-OPC module.

Initialization module includes the following main steps: (1) Layout fragmentation into segments and grouping each edge segments into clusters. (2) Dividing layout region into grids. (3) Generating initial mask solution M_0 whose RVKs intensity and local mask density are compared with their corresponding ones for target T to linearly interpolate the correction coefficient per grid.

6.2 Core-OPC Module

In Core-OPC module, a mask is iteratively adjusted through corner hammering, segment shifting, and alignment to achieve high mask manufacturability with preserving acceptable pattern fidelity under process variations. This module includes the following main steps: (1) Variational intensity error interpolation. (2) Two-Segment Shifting and Alignment. (3) Intensity Estimation for the adjusted mask using our proposed intensity estimation. (4) Mask evaluation in terms of EPE and PV-band. Once stop conditions are satisfied, the least cost mask is forwarded to the Post-OPC module. The following proposes our modeling for variational intensity error and Two-Segment shifting/alignment:

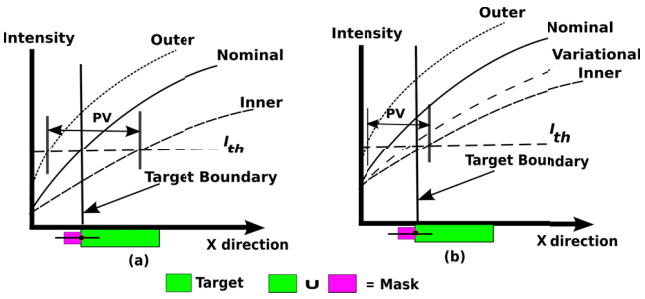


Fig. 12 (a) Inputting nominal intensity to OPC recipe with relatively high PV. (b) Inputting variational intensity to OPC recipe with relatively less PV due to nominal intensity slope increase at the tap point. The curves represent the intensities when going along the line perpendicular to the shown tap point on the boundary of T.

6.2.1 Variational Intensity Error Interpolation

Shifting a segment tries to make the intensity at the tap point of that segment equivalent to I_{th} as a response for the inputted intensity to minimize EPE. Nominal intensity $I_n(M,t)$ is inputted as the current estimated intensity of tap point t_i to the OPC recipe. However, this does not guarantee reducing the EPE under other process conditions, and thus, PV-band area would remain large.

In our algorithm, the intensity inputted to the OPC recipe is slightly tuned to include innermost, nominal, and outermost intensities. Such tuning aims to increase the nominal intensity slope at the tap point. Thus, we slightly make the inputted intensity less than I_n such that the OPC response makes an extra effort to make this tuned intensity (called variational intensity in this paper) equivalent to I_{th}. As a result, nominal intensity slope at the tap point will increase. When the slope increases, innermost intensity typically gets closer to I_{th} while outermost does not get so far from I_{th} since it is already saturated, therefore, the PV-band decreases. Figure 12(a) illustrates inputting I_n to the OPC recipe while Fig. 12(b) illustrates inputting the variational intensity, wherein, nominal intensity slope increases at the tap point of interest, and thus, PV-band gets smaller.

Variational intensity of a pixel p, denoted by $I_v(M,p)$, is a combination of innermost, nominal, and outermost intensity values in that pixel with weighting factors as given in Eq. (14) where: $\alpha \in [0,1]$, $\beta \in [0,1]$, $\alpha > \beta$, and $\alpha + 2 * \beta = 1.0$. Note that $\alpha > \beta$ to avoid large deviation of the nominal intensity from I_{th}. Additionally, the choice of β should guarantee that the variational intensity is slightly less than I_n following the given process window conditions. Variational intensity error, denoted by E_v, is the deviation of variational intensity of a pixel from I_{th}. It is interpolated per cluster as described in Sect. 5, but with considering variational intensity per tap point.

$$I_v(M,p) = \alpha * I_n(M,p) + \beta * (I_i(M,p) + I_o(M,p)) \quad (14)$$

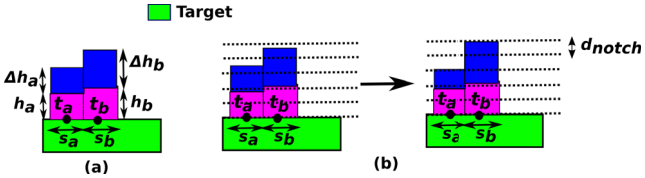


Fig. 13 (a) Two-Segment shifting. (b) Segment alignment.

6.2.2 Two-Segment Shifting and Alignment

Two-segment shifting proposed in [8] is adopted to minimize variational intensity error. Let s_a and s_b be two neighboring segments with shifting distances from the target boundary h_a and h_b, respectively, as shown in Fig. 13(a). The objective is to find their shifting distances $(\triangle h_a, \triangle h_b)$ from their current position that make the estimated variational intensity of their tap points equivalent to I_{th}. This problem is formulated in Eq. (15) where α_1 and α_2 are obtained by regression in [8].

$$\begin{bmatrix} I_v(M,t_a,h_a) + \alpha_1 \triangle h_a + \alpha_2 \triangle h_b \\ I_v(M,t_b,h_b) + \alpha_1 \triangle h_b + \alpha_2 \triangle h_a \end{bmatrix} = \begin{bmatrix} I_{\text{th}} + E_v(M,t_a,\triangle h_a) \\ I_{\text{th}} + E_v(M,t_b,\triangle h_b) \end{bmatrix} \quad (15)$$

To satisfy mask notch rule, a number of parallel lines to each target edge are created with d_{notch} spacing between consecutive lines. Once shifting distance of a segment is determined, this segment is aligned to the closest line parallel to it as shown in Fig. 13(b). Although such an alignment causes a tap point intensity to deviate from I_{th}, minimizing intensity error during shifting provides an acceptable margin for slight alignment under relaxed EPE evaluation.

6.3 Post-OPC Module

Post-OPC module aims to achieve higher mask manufacturability. This module includes: (1) EPE prediction modeling. (2) Two-Segment Concatenation. (3) Feature Shifting.

6.3.1 EPE Prediction Modeling

Let s_i be a segment with epe value of $epe(M,t_i)$ in its tap point t_i. With shifting s_i by distance $\triangle h_i$ from its position, the predicted EPE value induced by this shifting, denoted by $epe_{\text{pre}}(M,t_i,\triangle h_i)$, is given in Eq. (16) where $\beta(t_i)$ is called the EPE sensitivity factor of t_i.

$$epe_{\text{pre}}(M,t_i,\triangle h_i) \approx epe(M,t_i) + \beta(t_i) * \triangle h_i \quad (16)$$

6.3.2 Two-Segment Concatenation

The purpose of two-segment concatenation is to resolve mask notch rule violation in addition to reduce mask data volume. However, ad-hoc concatenation of neighboring segments impacts pattern fidelity.

Let s_a and s_b be two neighboring segments with $\triangle h_{ab}$

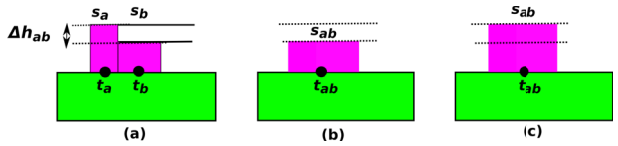


Fig. 14 (a) Before concatenation. (b) s_a is moving. (b) s_b is moving.

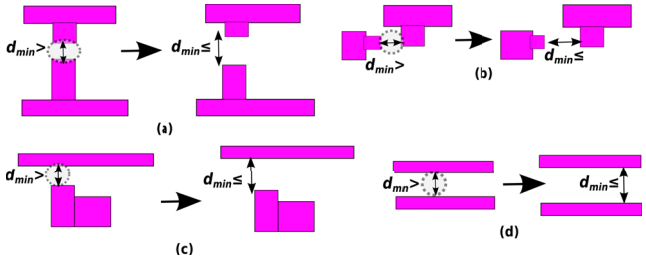


Fig. 15 Spacing violation resolution cases: (a) Two parallel features. (b) Two orthogonal features. (c) SRAF and segment. (d) Two SRAFs.

distance between them (Fig. 14(a)). Let epe_{pre_a} and epe_{pre_b} denote $epe_{\text{pre}}(M, t_a, \triangle h_{ab})$ and $epe_{\text{pre}}(M, t_b, \triangle h_{ab})$, respectively. Two-segment concatenation is applied as follows:

- If $epe_{\text{pre}_a} < epe_{\text{pre}_b}$ and $epe_{\text{pre}_a} \leq epe_{\max}$, shift s_a to concatenate with s_b (Fig. 14(b)).
- if $epe_{\text{pre}_b} < epe_{\text{pre}_a}$ and $epe_{\text{pre}_b} \leq epe_{\max}$, shift s_b to concatenate with s_a (Fig. 14(c)).
- If the predicted EPE causes violation in all the above mentioned cases, no concatenation is performed.
- If concatenation is done, s_a and s_b become one segment s_{ab} whose tap point t_{ab} is reassigned.

6.3.3 Feature Shifting

Feature shifting aims to resolve spacing violation between two features. For a comparison pair containing two parallel segments with spacing $d < d_{\min}$, both are shifted by $(d_{\min} - d)/2$ far from each others (Fig. 15(a)). The constraint to apply such shifting is that the predicted EPE in the tap points of both segments is beyond $epe_{\max}$. If the comparison pair contains two orthogonal features, the segment parallel to the target edge is shifted far from the other by $(d_{\min} - d)$, as long as no EPE violation is predicted (Fig. 15(b)).

If the comparison pair contains an SRAF, it is shifted far from the other feature to resolve spacing violation (Fig. 15(c)). If both features are SRAFs, they move equally far from each others (Fig. 15(d)). SRAF shifting is allowable as long as no additional EPE violations occur nearby.

6.4 Detailed OPC Algorithm

Our OPC algorithm is shown in Algorithm 1. The input is target T, minimum allowable mask notch d_{notch}, minimum allowable spacing $d_{\min}$, maximum allowable EPE $epe_{\max}$, set of RIKs and RVKs, and region invariant correction coefficient $\gamma_{\text{ri}}(K_{\text{ri}})$ obtained during preprocessing.

In initialization module, T is fragmented into set of segments where segment length L_{seg} is user-defined. The center of each segment is defined as tap point. For each edge, segments are grouped into clusters where cluster length is user-defined. Intensity map of T is simulated under various process conditions using top weight kernel k_0 and K_{rv} to obtain $I_x(T, \{k_0\})$ and $I_x(T, K_{\text{rv}})$, respectively, where $x \in \{i, n, o\}$.

Layout region is divided into set of grids W with predefined grid size. For each grid w_i, target density $\delta(T, w_i)$ and average region variant nominal intensity, $\widehat{I_n}(T, K_{\text{rv}}, w_i)$ is calculated. SRAFs are inserted, corners are hammered, and segments are shifted using approaches proposed in [8] to generate mask M_0 whose intensity map is simulated using k_0, K_{rv}, and K_{ri} kernels. For each grid w_i, $\gamma_{\text{rv}}(K_{\text{rv}}, w_i)$ is linearly interpolated as given in Eq. (11).

Core-OPC module consists of a number of iterations. Each iteration starts with calculating variational intensity map following Eq. (14). This map guides corner hammering and Two-segment shifting for each pair of neighboring seg-

Algorithm 1: Proposed OPC Algorithm

```
********** Initialization Module**********
Seg ← fragEdges(T, L_seg)
Tap ← assignTapPoints(Seg)
Clus ← clusterSeg(Seg, T)
for each x ∈ {i, n, o} do
    I_x(T, {k_0}) ← sim(T, {k_0}, x)
    I_x(T, K_rv) ← sim(T, k_rv, x)
    I_x(T) ← I_x(T, {k_0}) + I_x(T, K_rv, x)
end for
W ← divideRegionIntoGrids(R)
for each grid w_i ∈ W do
    δ(M, w_i) ← calculateMaskPixelsPerGrid(T, w_i)
    Î_n(T, K_rv, w_i) ← calculateAverageIntensity(w_i, I_n(M, K_rv))
end for
M_0 ← insertSRAFs(T, I_i(T), I_o(T), I_n(T)) //using algorithm described in [8]
M_0 ← hammerCorners(T, I_n(T)) //using algorithm described in [8]
M_0 ← TwoSegmentShift(T, I_n(T))
for each x ∈ {i, n, o} do
    I_x(M_0, {k_0}) ← sim(M_0, {k_0}, x)
    I_x(M_0, K_rv) ← sim(M_0, k_rv, x)
    I_x(M_0, K_ri) ← sim(M_0, k_ri, x)
    I_x(M_0) ← I_x(M_0, {k_0}) + I_x(M, K_ri) + I_x(M_i, K_rv)
end for
for each grid w_i ∈ W do
    δ(M_0, w_i) ← calculateMaskPixelsPerGrid(M, w_i)
    Î_n(M_0, K_rv, w_i) ← calculateAverageIntensity(w_i, I_n(M_0, K_ov))
    γ_rv(w_i) ← interpolate(δ(M_0, w_i), δ(T, w_i), Î_n(T, K_rv, w_i), Î_n(M_0, K_rv, w_i))
end for
I_v(M_0) ← α * I_n(M_i) + β * (I_i(M) + I_o(M_i))
E_v[Clus] ← 0
G_n(M_0) ← applyResistModel(I(M_0), I_th)
#EPEV ← calculateEPE(T, G_n(M_0))
i ← 1
********** Core-OPC Module**********
while #EPEV(M) > 0 AND i < maxIterNum do
    M_i ← hammerCorners(M_{i-1}, I_v(M_{i-1}))
    M_i ← TwoSegmentShift(M_{i-1}, I_v(M_{i-1}), Clus, E_v[Clus])
    M_i ← alignSeg(M_{i-1}, Seg)
    for each x ∈ {i, n, o} do
        I_x(M_i) ← findAdaptiveIntensityMap(M_i, M_{i-1}, W, γ_ri, γ_rv, I(M_i))
        G_x(M_i) ← applyResistModel(I(M_i), I_th)
    end for
    I_v(M_i) ← α * I_n(M_i) + β * (I_i(M) + I_o(M_i))
    E_v[Clus] ← interpolateVariationalErrorPerCluster(Clus, E_v[Clus], I_v(M_i))
    #EPEV ← calculateEPE(T, G_n(M_i))
    PV ← XOR(G_i(M_i), G_0(M_i))
    cost[M_i] ← A * #EPEV + B * PV
    i ← i + 1
end while
M ← chooseMaskWithLeastEPE(M, cost)
********** Post OPC Module**********
I_n(M) ← findAdaptiveIntensityMap(M, M, W, γ_ri, γ_rv, I(M))
Sen ← calculateEPESensitivity(M, T)
M* ← concatenateSegments(M, I_n(M), Sen, d_notch)
I(M*) ← findAdaptiveIntensityMap(M*, M, W, γ_ri, γ_rv, I(M))
Sen ← calculateEPESensitivity(M*, T)
M ← shiftFeatures(M*, I(M*), d_min)
I(M) ← findAdaptiveIntensityMap(M, M, W, γ_ri, γ_rv, I(M))
```

Table 1 Intensity estimation evaluation.

Benchmark	Static IDM with $M_{ref} = T$				Static IDM with $M_{ref} = M_0$				Adaptive IDM			
	#EPEV	PVBand	Time	Cost	#EPEV	PVBand	Time	Cost	#EPEV	PVBand	Time	Cost
B1	8	61074	64	284296	7	62548	67	285192	5	63033	65	277132
B2	0	50276	62	201104	0	50184	65	200736	0	49808	67	199232
B3	48	127556	66	750224	48	119252	70	717008	42	109250	73	647000
B4	2	25997	61	113988	2	24600	71	108400	2	24753	72	109012
B5	6	68505	68	304020	3	59871	71	244484	1	58312	81	238248
B6	1	51363	65	210452	2	48265	70	208060	0	51230	60	204920
B7	0	45121	63	180484	0	42852	59	171408	0	41687	62	166748
B8	0	23413	58	93652	0	21552	70	86208	1	23387	71	98548
B9	0	65887	67	263548	1	63304	69	258216	0	64841	62	259364
B10	0	17346	55	69384	0	17014	54	68056	0	17046	50	68184
Ratio			**0.95**	**1.09**			**1.00**	**1.04**			**1.00**	**1.00**

Time unit:Sec, PVBand unit: nm^2, Segment length= 20 nm, M_0: Initial OPCed mask

ments by solving Eq. (15) followed by segment alignment. Intensity map of the adjusted mask M_i is simulated using k_0 to obtain $I_x(M_i, \{k_0\})$ for which corrected $I_x(M_i, K_{ri})$ and $I_x(M_i, K_{rv})$ maps are added where $x \in \{i, n, o\}$. Those maps are corrected per grid using correction coefficients as described in Algorithm 2. The total intensity map $I_x(M_i)$ is used to interpolate the variational intensity error per cluster to guide next iteration adjustments.

$G_n(M)$ is extracted from $I_n(M)$ using CTR model. This image is used to calculate $\#EPEV$. Similarly, $G_i(M)$ and $G_o(M)$ are extracted from I_i and I_o, respectively, to calculate $PV(M)$. Mask cost is calculated using Eq. (17) where A and B are user-defined. Once $\#EPEV$ reaches 0 or the maximum number of iterations is exceeded, least cost mask is forwarded to Post-OPC module.

$$Cost(M) = A * \#EPEV(M) + B * PV(M) \tag{17}$$

In Post-OPC, EPE sensitivity factor is calculated for each tap point as given in Eq. (18) where h_i is the shifting distance of segment s_i from its origin in T. This factor guides Two-Segment concatenation stage in which neighboring segments are concatenated if no EPE violation is predicted. Intensity map is simulated using adaptive IDM approach and M undergoes feature shifting to resolve spacing violations. The output is the final mask solution.

$$\beta(t_i) = (epe(M, t_i) - epe(T, t_i))/h_i \tag{18}$$

Algorithm 2:findAdaptiveIntensityMap(M, M_{ref}, W, γ_{ri}, γ_{rv}, $I(M)$)
$I_1(M) \leftarrow$ findIntensityMapUsingTopWeightKernel(M)
for each grid $w_i \in W$ **do**
  $\delta(M, w_i) \leftarrow$ calculateMaskPixelsPerGrid(M, w_i)
  $\triangle\delta \leftarrow \delta(M, w_i) - \delta(M_{ref}, w_i)$
  for each pixel $p \in W_i$ **do**
    $I(M, K_{ri}, p) \leftarrow I(M_{ref}, K_{ri}, p) + \triangle\delta * \gamma_{ri}$
    $I(M, K_{rv}, p) \leftarrow I(M_{ref}, K_{rv}, p) + \triangle\delta * \gamma_{rv}(K_{rv}, w_i)$
    $I_1^*(M, p) \leftarrow I_1(M, p) + I(M, K_{ri}, p) + I(M, K_{rv}, p)$
  end for
end for
$I(M) \leftarrow I_1^*(M)$

7. Experimental Results

7.1 Experimental Setup

Our OPC algorithm is implemented on top of Lithosim simulator from ICCAD 2013 CAD contest [19] using C language on a 4 cores 3.6 GHz Linux machine with total memory of 1986912 KB. In Lithosim, layout region area is

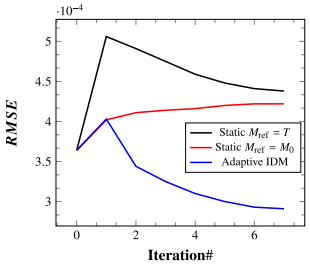


Fig. 16 RMSE of intensity change with OPC iterations for B6 using different estimation models.

1024×1024 pixels where each pixel represents 1 $nm \times 1$ nm. $K = \{k_0, k_1, \ldots, k_{23}\}$ is the set of kernels where k_0 is the top weight kernel. Nominal process condition is defined at (1.0 dose, best focus). Innermost and outermost conditions are defined at (0.98 dose, defocus) and (1.02 dose, best focus), respectively. In the contest, $epe_{max} = 15\,nm$, $I_{th} = 0.225$, mask cost function is $5000 * \#EPEV + 4 * PV$ where $\#EPEV$ and PV are calculated using contest checker tools [19].

In experiments, grid size is chosen $8\,nm \times 8\,nm$, $L_{seg} = 20\,nm$, $d_{notch} = 5\,nm$, $d_{min} = 10\,nm$, cluster length is 1/4 of the edge length, maximum number of iterations is 10. Top 5 weight kernels are used where $K_{rv} = \{k_1, k_2\}$ and $K_{ri} = \{k_3, k_4\}$. $G_o(M)$ is extracted from $I_n(M)$ using dose variation modeling published in [8]. For variational intensity, we use $\alpha = 0.5$, $\beta = 0.25$ for Eq. (14).

7.2 Algorithm Experimental Results

All experiments were executed on the public benchmarks released by IBM for ICCAD 2013 CAD contest [19].

7.2.1 Intensity Estimation Evaluation

Algorithm proposed in [8] was executed on the public benchmarks using: (1) Static IDM with $M_{ref} = T$. (2) Static IDM with initial OPCed mask as M_{ref}. (3) Adaptive IDM. For each benchmark, $\#EPEV$, PV, cost, and computation time are shown in Table 1. Adaptive IDM to estimate intensity outperforms others in terms of the cost related to EPE and PV-band area. This is a result of the improved accuracy in intensity estimation. Notice that the algorithm published in [8] with static IDM when $M_{ref} = T$ has around 5% reduction in the computation time if compared with our proposed algorithm. This is reasonable since [8] does not include one extra OPC step to generate the initial mask solution required

Table 2 Entire Algorithm Evaluation VS Algorithm published in [8] With Adaptive IDM.

Benchmark	Algorithm Published in [8] With Adaptive IDM							Our Proposed Algorithm						
	#EPEV	PVBand	Cost	#NotchV	#SpaceV	Volume	Time	#EPEV	PVBand	Cost	#NotchV	#SpaceV	Volume	Time
B1	5	63033	277132	166	28	7484	65	5	61657	271628	0	10	6208	83
B2	0	49808	199232	140	27	6070	67	2	49248	206992	0	9	4818	78
B3	42	109250	647000	220	29	9144	73	52	90523	622092	0	16	7421	79
B4	2	24753	109012	55	10	2932	72	4	26454	125816	0	4	2376	80
B5	1	58312	238248	165	26	6541	81	3	60843	258372	0	9	5543	81
B6	0	51230	204920	140	29	7252	60	1	48961	200844	0	12	6155	86
B7	0	41687	166748	94	20	4905	62	0	40995	163980	0	4	3555	79
B8	1	23387	98548	56	11	3829	71	0	21893	87572	0	4	3226	75
B9	0	64841	259364	190	29	8518	62	0	64140	256560	0	10	6260	80
B10	0	17046	68184	59	0	2757	50	0	17240	68960	0	0	2329	69
Ratio	**0.76**	**1.04**	**1.00**	**-**	**2.68**	**1.24**	**0.83**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Time unit:Sec, PVBand unit: nm^2, Segment length= 20 nm, Volume: Bytes

Table 3 Comparison with Top 3 teams in ICCAD contest.

Benchmark	3rd				2nd				1st				Our Algorithm			
	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time
B1	2	70014	290056	273	6	57190	258760	482	0	65743	262972	606	5	61657	271628	83
B2	0	58927	235708	130	13	45776	248104	485	1	53335	218340	319	2	49248	206992	78
B3	35	106676	601704	305	39	90493	556972	487	25	143993	700972	294	52	90523	622092	79
B4	1	38401	158604	287	14	24276	167104	487	0	31654	126616	414	4	26454	125816	80
B5	4	69796	299184	210	16	55754	303016	489	0	65529	262116	262	3	60843	258372	81
B6	0	59315	237260	91	18	49059	286236	482	1	62164	253656	430	1	48961	200844	86
B7	8	56972	267888	353	8	43663	214652	482	0	51098	204392	395	0	40995	163980	79
B8	0	26106	104424	80	0	23810	95240	531	0	25802	103208	239	0	21893	87572	75
B9	12	78781	375124	409	15	62164	323656	569	2	74931	309724	284	0	64140	256560	80
B10	0	18579	74316	60	0	19585	78340	489	0	18433	73732	172	0	17240	68960	69
Ratio			**1.17**	**2.80**			**1.12**	**6.28**			**1.11**	**4.37**			**1.00**	**1.00**

PVBand unit:nm^2, Time unit: Sec, Cost=5000 $*$ $\#EPEV + 4 * PVBand$

Table 4 Comparison with recently published algorithms.

Benchmark	MOSAIC Fast [6]				MOSAIC Exact [6]				PVOPC [13]				Our Algorithm			
	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time	#EPEV	PVBand	Cost	Time
B1	6	58232	262928	318	9	56890	272560	1707	2	58269	243076	164	5	61657	271628	83
B2	10	47139	238556	256	4	48312	213248	1245	0	52674	210696	130	2	49248	206992	78
B3	59	82195	623780	321	52	84608	598432	2522	48	81541	566164	360	52	90523	622092	79
B4	1	28244	117976	322	3	24723	113892	1269	0	26960	107840	265	4	26454	125816	80
B5	6	56253	255012	315	2	56299	235196	2167	4	61820	267280	62	3	60843	258372	81
B6	1	50981	208924	314	1	49285	202140	2084	0	55090	220360	54	1	48961	200844	86
B7	0	46309	203924	239	0	46280	185120	1641	0	51977	207908	74	0	40995	163980	79
B8	2	22482	99928	258	2	22342	99368	663	0	22869	91476	65	0	21893	87572	75
B9	6	65331	291324	322	3	62529	265116	3022	0	70713	282852	55	0	64140	256560	80
B10	0	18868	75472	231	0	18141	72564	712	0	17846	71384	41	0	17240	68960	69
Ratio			**1.04**	**3.67**			**1.00**	**21.56**			**1.00**	**1.61**			**1.00**	**1.00**

PVBand unit:nm^2, Time unit: Sec, Cost=5000 $*$ $\#EPEV + 4 * PVBand$

to interpolate the correction coefficients. Nevertheless, the cost has been reduced by around 8% using our proposed algorithm.

Figure 16 illustrates Root Mean Square Error (RMSE) of nominal intensity (given in Eq. (19) where I^g is the golden intensity using all kernels, I^* is the estimated intensity, and Z is the set of tap points) change during OPC iterations which shows that with adaptive IDM, RMSE stays beyond RMSE of other approaches.

$$RMSE = \sqrt{\frac{1}{|Z|}\sum_{t\in Z}(I^g(M,t) - I^*(M,\{k_0\},t))^2} \quad (19)$$

7.2.2 OPC Algorithm Evaluation

Our proposed OPC algorithm and the algorithm published in [8] with being adopted to include adaptive IDM were executed on the public benchmarks. For each benchmark, $\#EPEV$, PV, Cost,$\#NotchV$, $\#SpaceV$, mask file size, and computation time are shown in Table 2.

Notch violations have been eliminated due to segment alignment and concatenation while almost 63% reduction has been achieved in spacing violations due to feature shifting. Mask data volume has been reduced by 20% due to Two-segment concatenation. With around 24% increase in EPE violations due to manufacturability consideration, PV-band area has been reduced by 4% due to variational intensity error consideration. Thus, overall, the cost in terms of EPE and PV-band is almost the same with [8] while the proposed algorithm effectively minimizes notch and spacing design rule violations. Computation time has increased by 17% due to the additional processing and imposing the post-OPC module.

7.2.3 Comparison with Recent Algorithms

Our results are compared with the top 3 teams in ICCAD 2013 contest and other recently published algorithms executed on the same benchmarks as recorded in Table 3 and Table 4, respectively. Our algorithm outperforms the top 3 teams in terms of EPE and PV-band area reduction with being almost 4X faster than the top team. The proposed algorithm has almost the same cost in terms of EPE and PV-band area with recent algorithms, but faster in terms of computation time due to exploiting IDM. Additionally, mask manufacturability is considered in our algorithm which results in less complex mask solutions.

Figure 17 illustrates a portion of the mask solution

Fig. 17 Portion of Benchmark B6 mask Outputted from: (a) MOSAIC Algorithm [6]. (b) Our Algorithm. Note that the mask in (a) includes many small notches which increases mask manufacturing cost in terms of mask data volume. The mask in (b) has more regularity which reduces its data volume due to segment concatenation and alignment.

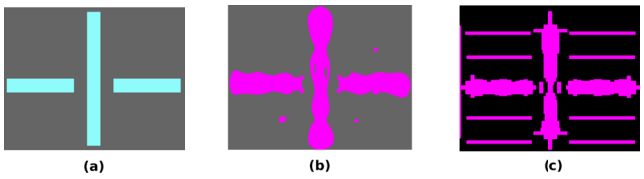


Fig. 18 Benchmark B4: (a) Target pattern. (b) Mask solution using MOSAIC algorithm [6]. (c) Mask solution using our proposed algorithm.

of benchmark B6 generated using MOSAIC algorithm (Fig. 17(a)) and our algorithm (Fig. 17(b)). With the pixel based behavior of MOSAIC algorithm, the mask shown in Fig. 17(a) contains many small notches while our generated mask solution looks more regular due to segment alignment and concatenation. The small notches in MOSAIC mask results in mask data volume explosion and such algorithm requires huge amount of data to store the coordinates of mask solutions for larger benchmarks. Furthermore, such small notches make a mask more sensitive to mask manufacturing process variations, which turns out into pattern fidelity degradation.

Figure 18(a) illustrates the target pattern for benchmark B4. Figure 18(b) and Fig. 18(c) illustrate the mask solutions generated using MOSAIC and our proposed algorithm. Figure 18(b) includes many small notches due to its pixel based behavior while Fig. 18(c) is generally more regular with satisfying mask notch design rule with small number of spacing rule violations as well.

8. Conclusions and Future Work

In this paper, we proposed a new algorithm to improve mask manufacturability with preserving acceptable pattern fidelity under process variations within short computation time. Adaptive intensity difference map helps in improving estimation accuracy without computation overhead. OPC steps modeling with variational intensity error plays a key role in creating a margin within which acceptable pattern fidelity is maintained with significant improvement in manufacturability. Experimental results show the effectiveness of our algorithm compared with existing work.

In the future, post OPC stage will be improved to consider hotspots detection and fixing in addition to line width design rule for better mask manufacturability.

Acknowledgement

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

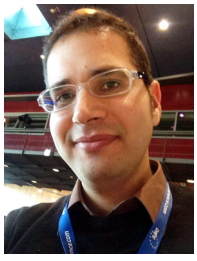
References

[1] X. Ma and G.R. Arce, Computational Lithography, Wiley Publisher, 2010.
[2] R.F. Pease and S.Y. Chou, "Lithography and other patterning techniques for future electronics," Proc. IEEE, vol.96, no.2, pp.248–270, 2008.
[3] N.B. Cobb and Y. Granik, "Model-based OPC using the MEEF matrix," Proc. 22nd Annual BACUS Symposium on Photomask Technology, pp.1281–1292, 2002.
[4] P. Yu, S.X. Shi, and D.Z. Pan, "Process variation aware OPC with variational lithography modeling," Proc. 43rd Annual Conference on Design Automation, DAC'06, pp.785–790, 2006.
[5] S. Tanaka, H. Tanabe, S. Inoue, T. Kotani, K. Izuha, and I. Mori, "Impact of OPC aggressiveness on mask maufacturability," Proc. Photomask and Next-Generation Lithography Mask Technology X, vol.5130, 23, 2003.
[6] J.-R. Gao, X. Xu, B. Yu, and D.Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," Proc. 51st Annual Design Automation Conference on Design Automation Conference, DAC'14, pp.1–6, 2014.
[7] L. Pang, M. Hoga, Y. Liu, and D. Abrams, "Inverse lithography technology (ILT): What is the impact to the photomask industry?," Photomask and Next-Generation Lithography Mask Technology XIII, 62830X, 2006.
[8] A. Awad, A. Takahashi, S. Tanaka, and C. Kodama, "A fast process variation and pattern fidelity aware mask optimization algorithm," Proc. 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.238–245, 2014.
[9] N. Cobb, "Sum of coherent systems decomposition by SVD", Univ. of California at Berkeley, pp.1–7, Sept. 21, 1995,
[10] L. Capodieci, P. Gupta, A.B. Kahng, D. Sylvester, and J. Yang, "Toward a methodology for manufacturability-driven design rule exploration," Proc. 41st Annual Conference on Design Automation, DAC'04, pp.311–316, 2004.
[11] P. Gupta, J.T. Weed, P.M. Martin, A.B. Kahng, S. Muddu, S. Nakagawa, and C.-H. Park, "Modeling OPC complexity for design for manufacturability," Proc. 25th Annual BACUS Symposium on Photomask Technology, 59921W, 2005.
[12] Y. Shim, V.K. Singh, M.L. Rieger, J. Choi, J. Kim, B. Su, P. Zhang, and K.-Y. Kim, "Improvement on OPC completeness through pre-OPC hot spot detection and fix," Proc. Design for Manufacturability through Design-Process Integration II, 692513, 2008.
[13] Y.-H. Su, Y.-C. Huang, L.-C. Tsai, Y.-W. Chang, and S. Banerjee, "Fast lithographic mask optimization considering process variation," Prc. 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.230–237, 2014.
[14] M. Lavin, L. Leibman, S. Mansfield, M. Mukhejree, and Z. Zhao, "System and method of smoothing mask shapes for improved placement of sub-resolution assist features," US Patent 7261981, 2007.
[15] T. Yasuzato, "Mask data creation method," US Patent 7691543B2, 2010.
[16] S. Shioiri and H. Tanabe, "Fast optical proximity correction: Analytical method," Proc. Optical/Laser Microlithography VIII, pp.261–269, 1995.
[17] G. Gallatin, K. Lai, M. Mukhejree, and A. Rosenbluth, "Printability verification by progressive modeling accuracy," US Patent 7512927, 2009.
[18] A. Awad, A. Takahashi, and C. Kodama, "A fast manufacturability aware optical proximity correction (OPC) algorithm with adaptive

wafer image estimation," Proc. 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.49–54, 2016.

[19] S. Banerjee, Z. Li, and S.R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," Proc. 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.271–274, 2013.

[20] P. Yu and D.Z. Pan, "A novel intensity based optical proximity correction algorithm with speedup in lithography simulation," Proc. 2007 IEEE/ACM International Conference on Computer-Aided Design, pp.854–859, 2007.



Ahmed Awad received his Bachelor and master degrees from the department of computer systems engineering, Birzeit university, Palestine, in 2009 and 2011, respectively. He worked as a quality assurance engineer for CISCO systems from 2010–2013. He is currently pursuing his PhD degree in the department of communications and computer engineering in Tokyo Institute of Technology, Japan. His research interests include VLSI physical design, OPC, and low power VLSI testing.

Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He had been with the Tokyo Institute of Technology as a research associate from 1991 to 1997, and as an associate professor from 1997 to 2009 and from 2012 to 2015, and as a professor from 2015. He had been with the Osaka University as an associate professor from 2009 to 2012. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2002 to 2003. He is currently with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. He is currently a member of IEEE Circuits and Systems Society Board of Governors (BoG) and the chair of CEDA All Japan Joint Chapter. His research interests are in VLSI layout design and combinational algorithms. He is a member of IEEE, ACM, IEICE, and IPSJ.

Chikaaki Kodama received the B.E., M.E., and D.E. degrees in electronic and information engineering from Tokyo University of Agriculture and Technology, Tokyo, Japan, in 1999, 2001, and 2006, respectively. He was with Fujitsu Ltd., Kawasaki, Japan, from 2001 to 2003, where he worked on development of custom computer-aided design for SPARC64 processor. He is currently with Toshiba Corporation, Yokohama, Japan. His research interests include design for manufacturability, especially mask optimization for optical lithography and VLSI layout design. He is a senior member of IEEE.