IPSJ Transactions on System LSI Design Methodology Vol.14 2–10 (Feb. 2021)

[DOI: 10.2197/ipsjtsldm.14.2]

Invited Paper

Computational Lithography Using Machine Learning Models

Youngsoo Shin[1,a)]

Received: October 2, 2020

Abstract: Machine learning models have been applied to a wide range of computational lithography applications since around 2010. They provide higher modeling capability, so their application allows modeling of higher accuracy. Many applications which are computationally expensive can take advantage of machine learning models, since a well trained model provides a quick estimation of outcome. This tutorial reviews a number of such computational lithography applications that have been using machine learning models. They include mask optimization with OPC (optical proximity correction) and EPC (etch proximity correction), assist features insertion and their printability check, lithography modeling with optical model and resist model, test patterns, and hotspot detection and correction.

Keywords: computational lithography, optical proximity correction, assist features, lithography model, lithography test patterns, hotspot

1. Introduction

Figure 1 illustrates basic elements of optical lithography system. Illumination is modeled by partial coherence factor σ. In partially coherent imaging, which improves the minimum resolvable pitch [1] and is a preferred imaging method, the mask is illuminated by light traveling in various directions. The smaller σ is, the higher the degree of illumination coherence. Projection, or exposure, is represented by numerical aperture NA $= n \sin\theta$. A critical dimension, usually a half of minimum pitch, corresponds to a minimum feature size attainable by a particular technology:

$$\mathrm{CD} = k_1 \frac{\lambda}{\mathrm{NA}}, \tag{1}$$

where λ is a wavelength of light source, and k_1 (often called k1 factor) is a measure of illumination complexity and is given by

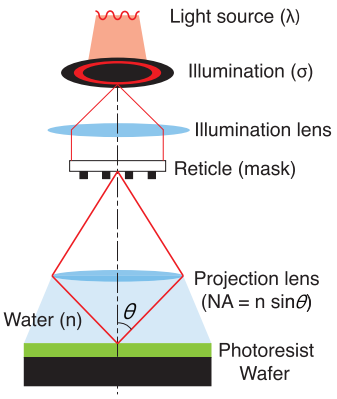

Fig. 1 Elements of optical lithography system.

$$k_1 = \frac{1}{2(1+\sigma)}. \tag{2}$$

For smaller CD, one obvious option is to advance to a light source of smaller wavelength, i.e., KrF with 248 nm to ArF with 193 nm, which is most popular now, to even more advanced EUV with 13.5 nm. Second option is to reduce k_1, which implies larger σ to be adopted. This is achieved through off-axis illumination (OAI), which involves the light source of annular, cross-pole, or quasar shapes. The last option is higher NA: this is through a medium with higher refractive index, e.g., water ($n = 1.33$) instead of air ($n = 1.0$), or projection lens of larger $\sin\theta$ value.

The k1 factor generally decreases with technology nodes [2]. The 500 nm and 350 nm nodes are imaged with $k_1 > 0.65$ with standard lithography. The 250 nm and 180 nm nodes approach k_1 of 0.5, and require the introduction of resolution enhancement techniques (RET). In 130 nm and 90 nm nodes, which are well below $k_1 = 0.5$, RETs are in widespread use. Theoretical limit of k_1 is 0.27; the nodes with k_1 smaller than that can only be imaged through multiple patterning technology.

1.1 Machine Learning for Computational Lithography

Computational lithography involves the use of computers to improve the resolution achievable through optical lithography. A key is lithography simulation, which is based on lithography models. A number of RET techniques have been introduced and used together with lithography simulation. They include optical proximity correction (OPC) adopted since 130 nm, subresolution assist feature (SRAF) since 90 nm, phase shift mask (PSM) and off-axis illumination (OAI) since 65 nm, double patterning since 20 nm, and directed self-assembly and triple patterning since 10 nm.

Computational lithography has relied on compact modeling for long. A resist model, for instance, captures development process,

[1] School of Electrical Engineering, KAIST, Daejeon 34141, Korea
[a)] youngsoo.shin@kaist.ac.kr

and is given by a weighted sum of convolutions between light intensity and Gaussians. Machine learning model offers higher modeling capability than simple polynomial, so its application provides compact modeling with higher accuracy.

Many lithography applications are computationally expensive, because they are iterative and often involve lithography simulations. A well trained machine learning model provides a quick estimation of outcome, and benefits these applications.

A number of lithography applications, which have been using machine learning models, are reviewed in this tutorial: mask optimization including OPC and EPC in Section 2, SRAF insertion and printability check in Section 3, lithography modeling in Section 4, test patterns in Section 5, and hotspot detection and correction in Section 6.

2. Mask Optimization

Figure 2 (a) shows a patterning process in optical lithography. Photomask pattern goes through a lithography process to form a pattern on photoresist, which is called resist- or aerial-image. Resist development and etch follow to finally form a wafer pattern.

Mask synthesis and optimization steps are illustrated in Fig. 2 (b), which can be considered as a reverse process of Fig. 2 (a). Ideally, the final wafer pattern should be the same as designers' layout. The goal of etch proximity correction (EPC), also called retargeting, is to synthesize an aerial image, which will yield the target wafer pattern even under non-ideal development and etch process. The goal of optical proximity correction (OPC) then is to synthesize a mask pattern, which produces the target aerial image (set by EPC) under light interference.

2.1 OPC

Most popular OPC method is model-based OPC (MB-OPC). It relies on iterative mask correction and lithography simulation. Each edge of initial mask pattern is divided into a number of segments (**Fig. 3** (a)), through a step called fragmentation. Each segment is individually moved by the amount called mask bias through correction step (Fig. 3 (b)). Lithography simulation follows to estimate the contour of aerial image, which is then compared to target aerial image at each fragmentation point (Fig. 3 (c)). The result is a number of EPE (edge placement error) values. MB-OPC is run with the given number of iterations of correction and simulation or until the target EPE (either maximum or average) is achieved.

MB-OPC is computationally expensive. With smaller feature size, it requires larger runtime due to more iterations to meet smaller EPE target and increased simulation time to model more complex light interference. The number of critical layers that should go through OPC also increases. The OPC runtime at 5 nm with 66 critical layers is 5.6× of runtime at 28 nm with 18 critical layers [3].

2.1.1 Fast OPC with ML Models

A number of OPC methods using machine learning models (ML-OPC) have been proposed to provide a quick OPC solution. The idea is illustrated in **Fig. 4**. A few features are extracted from the target segment to be corrected and its surroundings in the range of optical influence. The features are provided to the input layer of machine learning model, which has been trained beforehand. The values propagate through the hidden layers in the network until they reach the output layer. One node in the output layer with value 1 yields a predicted mask bias in case of classification, or a single node in the output layer may return mask bias value in regression model.

This approach is very fast, e.g., more than 10 times [4] faster than MB-OPC, because correction is done just once and no lithography simulations are performed. Accuracy, however, is limited even though the model is trained well, e.g., its maximum EPE is about 4 times larger than that of MB-OPC. For practical application, ML-OPC may be considered as a generator of initial OPC solution, which is provided to MB-OPC to deliver the final OPC result with just a few iterations. This hybrid approach is still faster than MB-OPC alone, e.g., about 3 times [4], and is being considered as an approach for commercial use [3].

Implementation Details: A key in ML-OPC implementation is a choice of features that should be extracted from a target segment. Discrete cosine transform signals have been used [5] as inputs of simple linear regression model. Local layout densities are popular features to represent a layout, which have been used with a hierarchical Bayesian model [6]. It has been shown that using po-

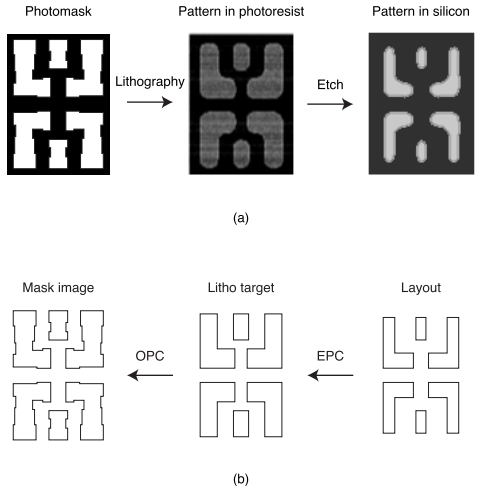

Fig. 2 (a) Photolithography process and (b) mask synthesis process.

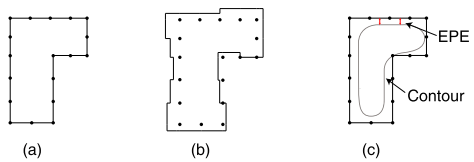

Fig. 3 OPC steps: (a) after fragmentation, (b) correction of mask pattern, and (c) lithography simulation result (Contour) compared to target pattern to calculate EPE.

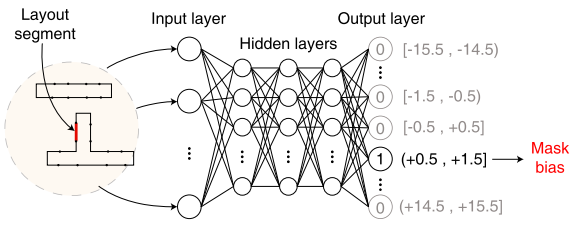

Fig. 4 ML-OPC using MLP (multilayer perceptron) model [4].

lar Fourier transform (PFT) signals can substantially reduce the number of features [7]. As will be shown in Section 4, a PFT signal, which is a convolution of PFT basis function (or optical kernel function) and local layout centered at target segment, is a component in light intensity calculation and thus well represents light interference around the segment.

A simple MLP, illustrated in Fig. 4, has been used as mask bias prediction model. More recently, a number of MLP instances that are connected through recurrent hidden layers, which is equivalent to recurrent neural network (RNN), are shown to provide higher accuracy [8]. This is because a target segment is corrected while its neighbor segments are corrected together in RNN, which better reflects the actual correction step (as in MB-OPC) as opposed to a simple MLP model where only a target segment is corrected.

Another issue in efficient implementation of ML-OPC is model training. Given a set of training segments with their reference mask bias values (through MB-OPC runs, for example), the goal is to train the model, i.e., determine the network structure and network parameters such as edge weights and node biases, such that the prediction of mask bias is as accurate as possible. A key in this process is sampling training segments, because using all segments from sample layouts is a waste of time and may cause overfitting of the model toward the segments which occur more frequently. Since sample layouts may not contain all segments that may arise in actual OPC process, generation of synthetic patterns, discussed in Section 5, may help extend the coverage of machine learning model.

2.2 EPC

During etch process, as shown in **Fig. 5**, some patterns experience over-etch due to photoresist erosion, which causes negative etch bias; some others are affected by under-etch due to excessive deposition, which causes positive etch bias. The goal of EPC is to modify a known target wafer pattern to compensate for etch biases, i.e., to synthesize an aerial image that can yield the designers' layout on the wafer even under over- and under-etch phenomena (see Fig. 2).

A key in EPC is etch bias model. A number of test patterns are created on a wafer, and etch bias is recorded for each pattern through CD measurement. The results may be summarized as a set of rules (in RB-EPC), e.g., etch bias table with line width and line spacing as parameters. The results may be fitted into a function of a few empirical parameters (in MB-EPC):

$$\text{Etch bias} = C_0 + C_1\text{Den} + C_2\text{Vis} + C_3\text{Blo} + C_4\text{Den}^2 + \cdots, \quad (3)$$

where Den is the density of the layout within a density kernel region; Vis is the area of the open space that is not hidden by the edges that are neighbors of a point of interest (i.e., space beyond the nearest edge is ignored); Blo is the area of the nearest polygon that overlaps with the blocked kernel, as shown in **Fig. 6** [10]. The coefficients C_i and the number of terms in this function are determined empirically through regression. MB-EPC can deal with a greater range of patterns than RB-EPC does, but it still fails to achieve a satisfactory on-chip variation (OCV). It is estimated that OCV in 20 nm DRAM devices is still 15% of gate size after MB-EPC has been applied [10].

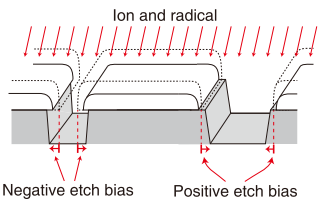

Fig. 5 Negative and positive etch bias [9].

2.2.1 EPC Using ML Models

Instead of Eq. (3), machine learning may be introduced to build an etch bias model. A simple MLP has been shown to offer much smaller RMS error of etch bias prediction [9]: 9.1 nm for rule-based, 2.9 nm for model-based, but 1.9 nm with MLP. A key is the choice of input parameters. Figure 6 indicates that local layout densities are important; in fact, they affect the quantity of etching particles, their incident angle and direction; they can be measured as illustrated in **Fig. 7**. Optical kernel signals are also important, since they affect the photoresist sidewall angle. Experiments indicate that using only layout densities for MLP causes 3.5 nm error in etch bias prediction, but including additional optical kernel signals brings the error down to 1.9 nm [9].

It is also noted that regression network, MLP with single output node to report etch bias, is better choice for smaller etch bias range when etch process is weak; classification network, MLP with multiple decision output nodes with each node associated with a small range of etch bias, is well suited for larger etch bias with strong etch process.

Once etch bias model is set up, actual correction can be performed through iteration as illustrated in **Fig. 8**. Some initial aerial image is assumed (L1), etch bias model is applied (L3)

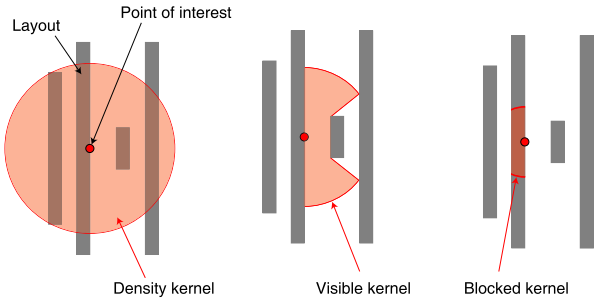

Fig. 6 Kernels for MB-EPC.

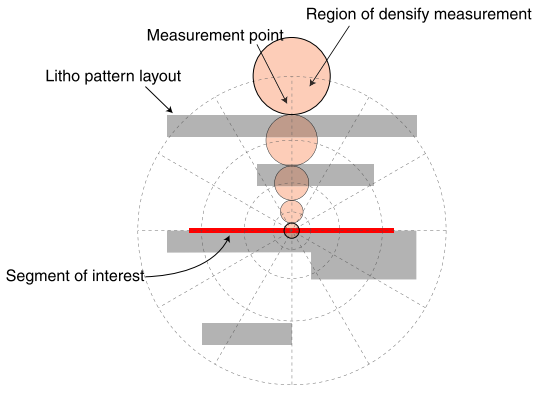

Fig. 7 Extraction of local layout densities.

Input: A design layout $\mathcal{D}_{\text{in}}$
Output: A litho pattern layout $\mathcal{L}_{\text{out}}$

L1: $\mathcal{L} \leftarrow \mathcal{D}_{\text{in}}$
L2: **repeat** for max_iterations
L3: A set of biases $\leftarrow ANN$(a set of segments from $\mathcal{L}$)
L4: $\mathcal{D} \leftarrow ETCH(\mathcal{L},$ a set of biases)
L5: A set of EPEs $\leftarrow Measure\_EPE(\mathcal{D}_{\text{in}}, \mathcal{D})$
L6: **if** $EPE_{\max} \leq \varepsilon$ **then** Exit loop
L7: $\mathcal{L} \leftarrow CORRECT(\mathcal{L}, -\alpha \times$ a set of EPEs)
L8: **return** $\mathcal{L}_{\text{out}} \leftarrow \mathcal{L}$

Fig. 8 Pseudo code of ML-EPC [9].

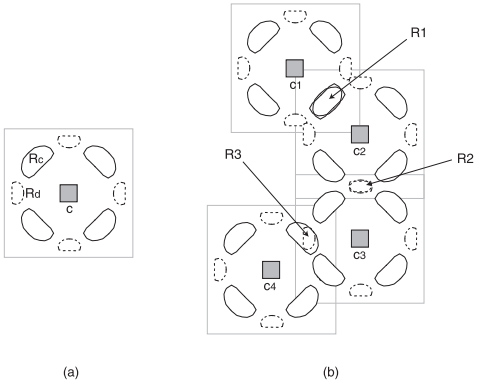

Fig. 9 LIM of (a) single contact and (b) a group of contacts [11].

to estimate wafer image (L4) which is compared to target image (L5), and the process is repeated with some (systematic) modification being applied to aerial image (L7).

3. Assist Features

Assist feature, or sub-resolution assist feature (SRAF) or scattering bar, is a key resolution enhancement technique (RET) in low k_1 lithography. SRAFs are extra patterns added to the mask, not intended to be printed on the wafer, which help nearby main patterns to be printed with higher fidelity and improved process window. Intuitively, sparse lines exhibit broader linewidth variations than dense lines, so adding assist features to both sides of sparse lines creates a dense environment [1].

SRAFs are usually inserted before OPC and are refined while main patterns are corrected through OPC. Rule-based methods (RB-SRAF) have been used since 90 nm node. A few empirical rules are established (e.g., the number of assist wires for each distance between adjacent metal wires) and are applied, either manually or automatically. They require long development time and are only limited to simple assist features. Model-based methods (MB-SRAF) have been widely used since 20 nm node. They rely on repeated SRAF insertion and its evaluation through lithography simulations; they are accurate but very time consuming. Inverse lithography technology (ILT) is more advanced method, which explicitly solves the problem of mask optimization; it is even more computationally expensive.

3.1 SRAF Insertion Using Guidance Map

A faster MB-SRAF has been proposed [11]. A key concept is a light interference map (LIM), where a layout is overlaid with an array of values indicating the potential amount of light interference. LIM of a single contact c is illustrated in **Fig. 9** (a). R_c is a region with larger values; if some patterns are introduced in that region, light intensity at c increases due to constructive interference. The opposite is true in R_d region, where values are smaller due to destructive interference. LIM in Fig. 9 (a) is obtained through repeated lithography simulations while some small pattern is moved around c. LIM of a number of contacts is simply a superposition of LIMs as shown in Fig. 9 (b).

LIM is convenient for SRAF insertion as illustrated in **Fig. 10** : (1) Binary versions of LIM are obtained by using different threshold values as shown in (b). If the value in the original LIM is larger than threshold, it is assumed 1 in binary LIM; it is assumed 0 otherwise. (2) Each binary LIM is discretized for easier SRAF insertion, as shown in (c). (3) One binary LIM (e.g., 5th one) is picked for initial SRAF insertion. (4) A lithography simulation is performed on the initial SRAF insertion, shown in (d), in which red contours are lithography images of contacts as well as SRAFs. (5) If an SRAF is associated with lithography image (its light intensity exceeds 100% of image threshold), it will be patterned; it is therefore replaced by its smaller version in binary LIM one level above. If the intensity of an SRAF is below 80% of image threshold (and so it is not associated with lithography image), it is replaced by its bigger version in binary LIM one level below. Refinement procedure (4) and (5) repeat until the intensity of all SRAFs lies between 80% to 100% of image threshold. The number of lithography simulations is 5 times at most, which is usually much smaller than the number associated with standard MB-SRAF.

SRAF guidance map (SGM) has been proposed for fast MB-SRAF [12]. Its concept is similar to LIM: SGM value indicates the sensitivity of improving process windows on the desired pattern.

3.2 SRAF Insertion Using ML Models

Deep CNN (convolutional neural network) has been applied for quick prediction of SRAF guidance map [13]. Runtime of SRAF insertion is reported to be reduced to 1/7 of standard insertion method using CTM (continuous transmission mask), which is conceptually similar to SGM.

The method has also been applied to ILT. Standard ILT involves 100 iterations of CTM refinement followed by 50 iterations of actual ILT process. Deep CNN allows a quick estimation of CTM without iterations, which is then followed by 50 iterations of ILT process. Runtime is reduced by about 34%. Data augmentation is important to extend the coverage of training data. Flipping, rotation, and translation are applied to initial training data for this purpose. Synthesis of test patterns covered in Section 5.2 is another possibility.

3.3 SRAF Printability Check

SRAFs are not intended to be printed on the wafer. Patterned SRAFs are considered as defects and become yield detractor. Printability check is thus an essential component of SRAF.

Lithography simulation is usually performed assuming bottom of resist height when main patterns are simulated. More pessimistic approach is necessary to simulate SRAF patterns, because miss prediction (predicted as non-printing for actually printed SRAFs) should be avoided more than false alarm (pre-

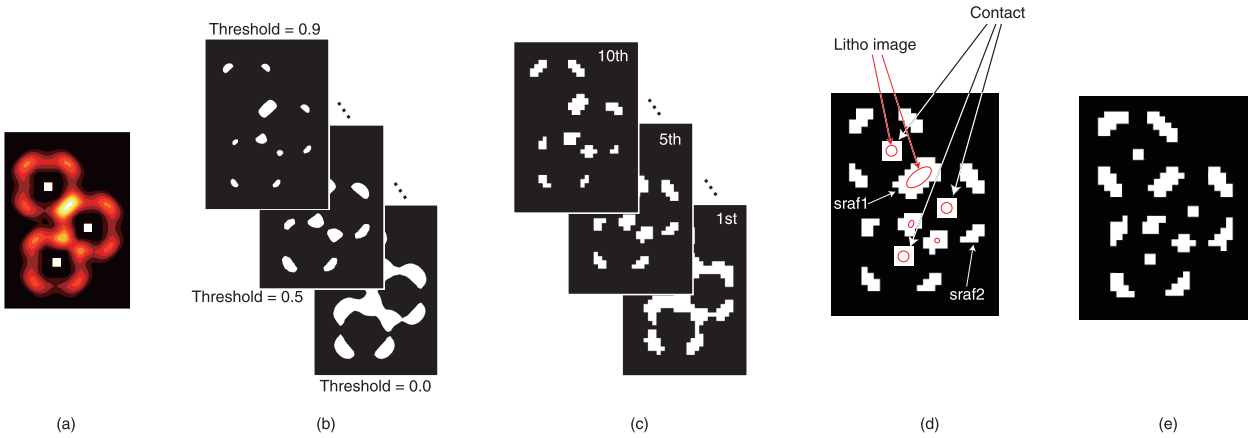

Fig. 10 SRAF placement using LIM: (a) initial LIM of a contact layout, (b) binary LIMs with different threshold, (c) discretized binary LIMs, (d) initial SRAF placement, and (e) refined final SRAF placement [11].

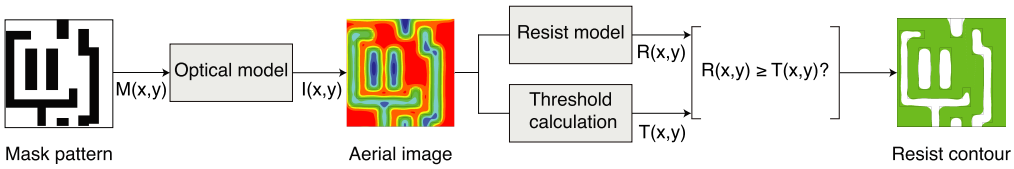

Fig. 11 Lithography model.

dicted as printing for actually non-printed SRAFs). One approach is to assume over-exposure condition while the same bottom of resist height is assumed for SRAFs; another is to assume the top or near the top of resist height with nominal exposure condition [14].

Machine learning approach has also been studied [15]. For each candidate pixel of SRAF, a machine learning model is applied to predict its printability. A simple MLP network has been tried with PFT signals and local layout densities (see Section 2) as network input parameters. Printability of SRAF is determined from the printability of its member pixels. Since SRAFs are not printed most of the time, balancing the number of reference printed SRAFs and non-printed SRAFs for model training is important.

4. Lithography Modeling

Lithography simulation is a foundation of computational lithography. It is based on lithography model, illustrated in **Fig. 11**, which describes the response of photoresist to exposure and development. Exposure is captured by optical model, in which image intensity is described by the weighted sum of convolutions between mask image $M(x, y)$ and optical kernel functions ϕ_i:

$$I(x, y) = \sum \lambda_i |\phi_i \otimes M(x, y)|^2 , \tag{4}$$

where λ_i is a weight value. This method is called sum of coherent systems (SOCS) approximation [16].

Development is captured by resist model, in which image intensity is modulated by resist process formula and the result is compared to some threshold values. Specifically, photoresist processing involves post-exposure bake (PEB) and resist development. The solution to differential equations for PEB modeling (or called reaction and diffusion), i.e., the quenching and diffusion of acid, is given by a sum of convolutions of $I(x, y)$ and Gaussians G_i [17]:

$$R(x, y) = c_0 I(x, y) + c_1 I(x, y) \otimes G_1(\sigma_1) + \cdots . \tag{5}$$

Development is modeled by comparing Eq. (5) to threshold value, either constant or variable. In popular variable threshold model [18], a threshold is given by a function of maximum intensity, minimum intensity, and intensity slope in local region of interest:

$$T(x, y) = C_0 + C_1 I_{max} + C_2 I_{min} + C_3 I_{slp} + C_4 I_{max}^2 + \cdots . \tag{6}$$

The weights in Eqs. (5) and (6) are determined through calibration process. Some test patterns are prepared, and calibration is performed such that the error of resist model is minimized, where the error may be measured as the difference of CD values between simulated resist patterns and corresponding patterns from actual measurement or from rigorous simulation. The choice of test patterns is thus very important, which is covered in Section 5.

4.1 ML for Resist Model

A number of options can be considered [19] in applying machine learning techniques in lithography model. A polynomial (6) for variable threshold may be replaced by machine learning model. CNN has been used [20] for this purpose. The input is an aerial image of a small clip and the output is corresponding intensity threshold. Experiments indicate that RMS error in predicted CD is about 5.5 nm with variable threshold model, while the method using CNN causes only 1.6 nm error.

A polynomial for resist model may be replaced by machine learning model [19]. Each convolution term in Eq. (5) becomes one input of machine learning model, and the output is $R(x, y)$ value. Experiments with 10 nm M1 layer demonstrates the drop

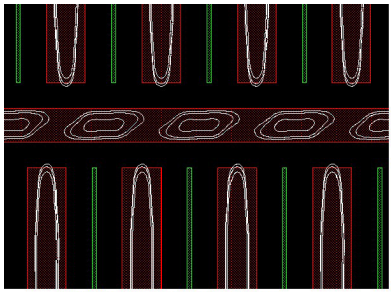

Fig. 12 Multiple resist contours at different resist heights.

of RMS error from 2.07 nm (when CM1 compact model [21] is used, in which threshold is constant) to 1.31 nm.

The resist model and the model for threshold in Fig. 11 may be put into a single machine learning model. A CNN has been used to predict CD from input aerial image [19], in which the accuracy is good but the parameters involved are too many and difficult to optimize. Another option is to use fully convolutional networks to predict resist contours from input aerial image [19]; in this application, reference resist contours (often extracted from SEM image) have to be very high quality [22].

4.1.1 Estimation of 3D Resist Profile

Standard resist model assumes two-dimensional space, but resist structure is often associated with non-vertical sidewalls. This may cause non-ideal resist profile, e.g., footing, T-topping, and top-less. Accurate prediction of 3D resist profile is therefore important. Rigorous simulation can be performed to predict 3D resist profile as illustrated in **Fig. 12**, but this is too time consuming.

A simple MLP network with local layout densities as shown in Fig. 7 and optical kernel signals as inputs produces accurate prediction of resist height [23], or a similar network may be trained to predict whether resist will remain after etch process.

4.2 ML for Lithography Model

A CVNN (complex valued neural network) has been applied for both optical- and resist-model [24]. The frequency components are limited, so small amount of training data can produce a model of higher accuracy even though there are extra processes of Fourier- and inverse Fourier-transform.

The CGAN (conditional generative adversarial network) model has been tried to obtain a wafer image directly from input mask image [25]. It has been applied to contact or via patterns, and extra CNN has also been applied to adjust the center of each pattern for higher accuracy.

5. Test Patterns

Comprehensive test patterns are important for a number of lithography applications including source mask optimization (SMO), building hotspot library, exploration of design rules, calibration of lithography models, and more recently training a machine learning model for lithography applications. Two types of test patterns are popular: parametric and actual. Parametric patterns are represented by a few geometrical parameters such as line width and space, as illustrated in **Fig. 13**. They are easy to build and analyze, but cannot cover complex patterns. Actual patterns are extracted from sample layouts and can cover more random

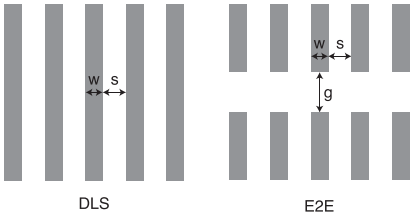

Fig. 13 Example parametric test patterns: dense line and space (DLS) and line-end to line-end (E2E).

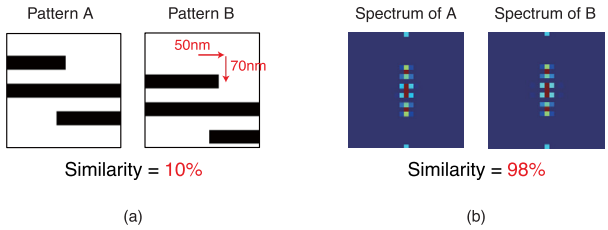

Fig. 14 (a) Area-based pattern matching and (b) matching in Fourier domain.

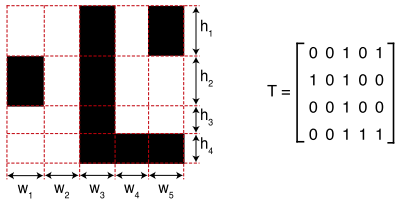

Fig. 15 Squish pattern representation of layout pattern.

shapes, but some similar shapes may be more frequent and some are not really important. Thus the extraction of important shapes, e.g., extracting hotspot patterns [26], and classifying them are important.

5.1 Classification

Once test patterns are assembled, they are classified into a small number of groups based on geometric similarity, or some other similarity measure of interest. One or more representative patterns is then identified from each group, and a collection of such representative patterns will form the final set of test patterns.

A classical method of classification is area-based pattern matching shown in **Fig. 14** (a); the percentage of overlap between the two patterns is used as a similarity measure. In some applications, not all shapes are equally important. For example of hotspot patterns, hotspot in the center is important, and shapes closer to the hotspot have a greater impact than those far apart from the hotspot. The pattern may be weighted by the square of the complex degree of coherence [27], $\mu(x, y)^2$, before pattern matching is performed. These strict pattern matching methods do not capture the similarity when one pattern is a shifted (or rotated or reflected) version of the other. This can be alleviated through pattern matching in Fourier domain [26]; the two patterns in Fig. 14 (a), which are similar by just 10%, are now very similar in frequency domain as illustrated in Fig. 14 (b).

A layout pattern, which is in Manhattan geometry, can be represented through Hanan grid, or called Squish pattern [28]. As shown in **Fig. 15**, scan lines are drawn from the extensions of all polygon edges, which divide the pattern into grids of non-regular interval. A binary matrix, in which 1 indicates the region occu-

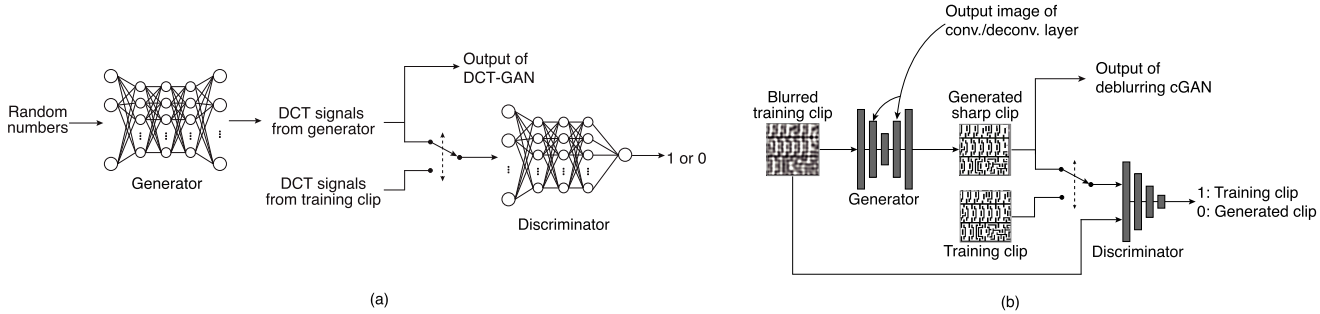

Fig. 16 Test pattern synthesis using GAN models: (a) GAN to generate DCT values and (b) CGAN to make blurred test pattern sharper.

pied by the pattern, together with a list of grid width and grid height information represents the pattern. The two patterns shall be similar if their corresponding matrices are the same but grid sizes are different within some tolerance range.

In many lithography applications, classification through strict pattern matching is inefficient and unnecessary. In lithography modeling, for instance, image parameter space (IPS) [29] consisting of intensity slope (I_{slp}) at pattern edge together with maximum-intensity (I_{max}) and minimum-intensity (I_{min}) in its close proximity is popular to define a parameter space. IPS may be extended to include more parameters to define a higher dimensional parameter space. e.g., IPS sensitivity to geometry perturbation [30] ($\partial I_{max}/\partial M$, where M is line width). Since the choice of parameter space is only engineering, alternative method is to introduce a machine learning to extract a number of parameters [31] and use them to define a parameter space. Once test patterns are identified in IPS space, any clustering algorithms can be applied for classification purpose: partitioning methods (such as K-means) or hierarchical methods (such as complete-link). The parameter space is also convenient for analysis of test pattern coverage.

5.2 Synthesis of Test Patterns

Pattern coverage through parametric patterns or actual patterns is always limited. This can be alleviated through automatic synthesis of test patterns. Machine learning approaches have been applied for this purpose.

Transforming auto encoder is a type of machine learning model suited for image translation, e.g., generating a shifted image [32]. Its key component, called capsule, consists of recognition units and generation units. A recognition unit is made of a few convolution layers and fully connected layers to identify the input image and generate a vector that characterizes the image, called latent vector. A generation unit is made of fully connected layers followed by deconvolution layers, with its function opposite to that of recognition unit. The idea of using transforming auto encoder is to systematically alter the latent vector so that the output pattern is slightly different from the input pattern [33]. The pattern is represented by Squish pattern, a matrix that identifies 2-dimensional topology as shown in Fig. 15, so pattern synthesis is realized by altering the matrix entries.

More general approach toward test pattern synthesis has been proposed [34]. A layout clip is represented by a few low frequency discrete cosine transform (DCT) signals. A layout often contains some repeated patterns, and DCT captures such repetition with a smaller number of frequency signals. Some random DCT signals are generated using GAN model as shown in **Fig. 16** (a); the block generator is trained beforehand such that it is difficult to discriminate its output from the DCT signals of clips used for training, i.e., DCT signals are generated such that they are realistic and close to the ones for training clips. The output DCT signals are provided to inverse DCT process to yield the corresponding clip image, which is blurred since only low frequency components are contained in the DCT signals. The clip is then made sharper by using CGAN model shown in Fig. 16 (b), which is trained such that it is also difficult to discriminate the sharpened clip from the ones that are used for training. The approach has been applied to resist modeling. When 1,000 parametric patterns are used for resist modeling, RMSE (root mean square error) of CD values is 5.11 nm; when they are replaced by 500 parametric patterns, 250 actual patterns, and 250 synthesized ones, CD RMSE becomes only 2.88 nm due to wider coverage of test patterns.

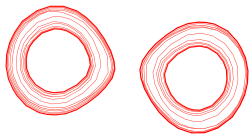

Fig. 17 PVBs of two contacts.

6. Hotspot

Hotspot patterns are the ones that may cause defects such as bridging, necking, and line-end shortening. They can be identified through process variation band (PVB). **Figure 17** shows an example. Lithography process is under the influence of key parameters: scanner focus, exposure energy, and mask manufacturing error. To account for parameter variations, lithography simulation may be repeated while each parameter is set to its mean or $\pm 3\sigma$ values. A set of resulting 27 contours is PVB. When two patterns are too close, their PVBs become thicker and so the minimum distance between the PVBs gets smaller as illustrated in Fig. 17. This may cause bridge. Hotspot may also be defined in probabilistic fashion [35].

Since repeated lithography simulations to get PVBs and detect hotspots are time consuming, a practical approach is to build a library of hotspot patterns beforehand through pattern classification and apply pattern matching to narrow down the region for actual lithography simulations.

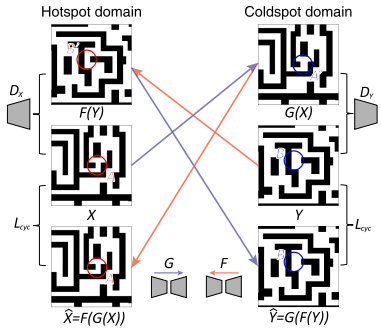

Fig. 18 Hotspot correction using cycleGAN [38].

6.1 ML for Hotspot Detection and Correction

Hotspot detection using CNN has been proposed [36]. Since hotspots are sparse, it is important to augment sample hotspot patterns so that CNN is well trained. A simple flipping and rotating have been tried for this purpose, even though they will not be enough for diverse patterns.

Automatic correction of hotspot using cycleGAN (cycle-consistent GAN [37]) has been proposed [38]. A set of sample hotspot patterns is X, and a set of coldspot patterns is Y as illustrated in **Fig. 18**. A hotspot pattern and coldspot pattern are not paired, i.e., Y does not necessarily contain the corrected version of hotspot from X, which is key advantage of using cycleGAN. The goal is to learn mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$, together with two discriminators D_X and D_Y. The objectives in learning process are: (1) G tries to generate a pattern $G(x \in X)$ that looks like a cold pattern in Y, against D_Y that aims to distinguish $G(x)$ from $y \in Y$ as much as possible. A similar objective is applied to F and D_X. (2) For each hotspot pattern x, G and F together should yield the pattern similar to x, i.e., $F(G(x)) \approx x$. Similarly we require $G(F(y)) \approx y$ for each cold pattern y. Experiments demonstrate an efficient correction of one or more various hotspots (tip-to-tip, tip-to-bar, pitch variation, density of neighbor patterns, etc.).

7. Conclusions

A number of computational lithography applications which employ machine learning models have been reviewed. Practical success so far is observed in OPC and lithography modeling. OPC using machine learning can provide a good initial OPC solution, which greatly helps reduce MB-OPC runtime. Lithography modeling has relied on empirical compact model for long; machine learning provides extensive modeling capability, which helps improve model accuracy.

SRAFs (insertion and printability check) and hotspot patterns (detection and correction) are also popular applications of machine learning. Preparation of sample training data is a challenge in these applications. Printed SRAF samples are obtained through scanning electron microscope (SEM) images, which should be carefully captured, measured, and classified [39]. Hotspot patterns as well as printed SRAF samples are often scarce, even though the success of machine learning model heavily relies on the coverage of training samples. Test patterns (extraction, classification, and synthesis) are thus important topic and need more study and development.

Acknowledgments This work was partly supported by the National Research Foundation of Korea (NRF) of Ministry of Science and ICT (MSIT) (No.2019R1A2C2003402), and SK Hynix. The author would like to acknowledge the help from Mr. Gangmin Cho, Mr. Pervaiz Kareem, and Mr. Yonghwi Kwon, all from School of Electrical Engineering, KAIST, Korea.

References

[1] Wong, A.: *Resolution Enhancement Techniques in Optical Lithography*, SPIE Press (2001).
[2] Liebmann, L.: Layout impact of resolution enhancement techniques: Impediment or opportunity?, *Proc. Int. Symp. Physical Design*, pp.110–117 (2003).
[3] Cao, Y.: Applications of machine learning in computational lithography, at SPIE eBeam lunch (2019).
[4] Choi, S., Shim, S. and Shin, Y.: Neural network classifier-based OPC with imbalanced training data, *IEEE Trans. CAD*, Vol.38, No.5, pp.938–948 (2018).
[5] Gu, A. and Zakhor, A.: Optical proximity correction with linear regression, *IEEE Trans. Semiconductor Manufacturing*, Vol.21, No.2, pp.263–271 (2008).
[6] Matsunawa, T., Yu, B. and Pan, D.Z.: Optical proximity correction with hierarchical Bayes model, *J. Micro/Nanolithography, MEMS, and MOEMS*, Vol.15, No.2, pp.021009-1–021009-8 (2016).
[7] Choi, S., Shim, S. and Shin, Y.: Machine learning (ML)-guided OPC using basis functions of polar Fourier transform, *Proc. SPIE 9780, Optical Microlithography XXIX* (2016).
[8] Kwon, Y., Song, Y. and Shin, Y.: Optical proximity correction using bidirectional recurrent neural network (BRNN), *Proc. SPIE 10962, Design-Process-Technology Co-optimization for Manufacturability XIII*, 109620D (2019).
[9] Shim, S. and Shin, Y.: Machine learning-guided etch proximity correction, *IEEE Trans. Semiconductor Manufacturing*, Vol.30, No.1, pp.1–7 (2017).
[10] Park, J., Kim, S., Shim, S. and Oh, H.: The Effective Etch Process Proximity Correction Methodology for Improving on Chip CD Variation in 20nm node DRAM gate, *Proc. SPIE Advanced Lithography*, pp.791–797 (2011).
[11] Shim, S., Choi, S. and Shin, Y.: Light interference map: A prescriptive optimization of lithography-friendly layout, *IEEE Trans. Semiconductor Manufacturing*, Vol.29, No.1, pp.44–49 (2016).
[12] Tsai, M. et al.: A full chip MB-SRAF placement using the SRAF guidance map, *Proc. SPIE 7823, Photomask Technology 2010*, 78233Q (2010).
[13] Wang, S. et al.: Machine learning assisted SRAF placement for full chip, *Proc. SPIE 10451, Photomask Technology 2017*, Vol.10451, pp.95–101 (2017).
[14] Viswanathan, R., Azpiroz, J. and Selvam, P.: Process optimization through model based SRAF printing prediction, *Proc. SPIE 8326, Optical Microlithography XXV*, 83261A (2012).
[15] Kwon, Y., Yang, J., Kim, S., Kim, C. and Shin, Y.: SRAF printing prediction using artificial neural network, *Proc. SPIE 11327, Optical Microlithography XXXIII*, 113270C (2020).
[16] Cobb, N., Zakhor, A. and Miloslavsky, E.: Mathematical and CAD framework for proximity correction, *Proc. SPIE 2726, Optical Microlithography IX* (1996).
[17] Zuniga, C., Deng, Y. and Granik, Y.: Resist profile modeling with compact resist model, *Proc. SPIE 9426, Optical Microlithography XXVIII*, 94261R (2015).
[18] Randall, J. et al.: Variable-threshold resist models for lithography simulation, *Proc. SPIE 3679, Optical Microlithography XII* (1999).
[19] Adam, K. et al.: Using machine learning in the physical modeling of lithographic processes, *Proc. SPIE 10962, Design-Process-Technology Co-optimization for Manufacturability XIII*, 109620F, pp.86–93 (2019).
[20] Watanabe, Y., Kimura, T., Matsunawa, T. and Nojima, S.: Accurate lithography simulation model based on convolutional neural networks, *Photomask Japan 2017: XXIV Symposium on Photomask and Next-Generation Lithography Mask Technology*, Takehisa, K. (Ed.), Vol.10454, pp.101–109, SPIE (2017).
[21] Granik, Y., Medvedev, D. and Cobb, N.: Toward standard process models for OPC, *Proc. SPIE 6520, Optical Microlithography XX*, 652043, pp.1447–1452 (2007).
[22] Kim, Y. et al.: OPC model accuracy study using high volume contour based gauges and deep learning on memory device, *Proc. SPIE 10959, Metrology, Inspection, and Process Control for Microlithography XXXIII*, 1095913 (2019).

[23] Shim, S., Choi, S. and Shin, Y.: Machine learning-based 3D resist model, *Proc. SPIE 10147, Optical Microlithography XXX*, 101471D (2017).
[24] Lee, H. et al.: A physics-driven complex valued neural network (CVNN) model for lithographic analysis, *Proc. SPIE 11326, Advances in Patterning Materials and Processes XXXVII* (2020).
[25] Ye, W. et al.: LithoGAN: End-to-end lithography modeling with generative adversarial networks, *Proc. Design Automation Conf.*, pp.1–6 (2019).
[26] Shim, S. and Shin, Y.: Topology-oriented pattern extraction and classification for synthesizing lithography test patterns, *J. Micro/Nanolithography, MEMS, and MOEMS*, Vol.14, No.1, pp.013503-1–013503-12 (2015).
[27] Ma, N. et al.: Automatic hotspot classification using pattern-based clustering, *Proc. SPIE 6925, Design for Manufacturability through Design-Process Integration II*, 692505, pp.35–44 (2008).
[28] Teoh, E. et al.: Systematic data mining using a pattern database to accelerate yield ramp, *Proc. SPIE 9053, Design-Process-Technology Co-optimization for Manufacturability VIII* (2014).
[29] Roessler, T., Frankowsky, B. and Toublan, O.: Improvement of empirical OPC model robustness using full-chip aerial image analysis, *Proc. SPIE 5256, 23rd Annual BACUS Symp. Photomask Technology*, pp.222–229 (2003).
[30] Vengertsev, D. et al.: The new test pattern selection method for OPC model calibration, based on the process of clustering in a hybrid space, *Proc. SPIE 8522, Photomask Technology 2012*, 85221A, pp.387–394 (2012).
[31] Chen, K. et al.: Full-chip application of machine learning SRAFs on DRAM case using auto pattern selection, *Proc. SPIE 10961, Optical Microlithography XXXII*, 1096108 (2019).
[32] Hinton, G., Krizhevsky, A. and Wang, S.: Transforming auto-encoders, *Proc. ICANN* (2011).
[33] Yang, H. et al.: DeePattern: Layout pattern generation with transforming convolutional auto-encoder, *Proc. Design Automation Conference*, ACM (2019).
[34] Kareem, P., Kwon, Y. and Shin, Y.: Layout pattern synthesis for lithography optimizations, *IEEE Trans. Semiconductor Manufacturing*, Vol.33, No.2, pp.283–290 (2020).
[35] Shim, S., Chung, W. and Shin, Y.: Lithography defect probability and its application to physical design optimization, *IEEE Trans. VLSI*, Vol.25, No.1, pp.271–285 (2016).
[36] Yang, H. et al.: Imbalance aware lithography hotspot detection: A deep learning approach, *Proc. SPIE 10148, Design-Process-Technology Co-optimization for Manufacturability XI*, 1014807 (2017).
[37] Zhu, J., Park, T., Isola, P. and Efros, A.: Unpaired image-to-image translation using cycle-consistent adversarial networks, *Proc. Int. Conf. Computer Vision* (2017).
[38] Sim, W. et al.: Automatic correction of lithography hotspots with a deep generative model, *Proc. SPIE 10961, Optical Microlithography XXXII*, 1096105 (2019).
[39] Kohli, K., Jobes, M. and Graur, I.: Automated detection and classification of printing sub-resolution assist features using machine learning algorithms, *Proc. SPIE 10147, Optical Microlithography XXX*, 1014700 (2017).

Youngsoo Shin received his B.S., M.S., and Ph.D. degrees in electronics engineering from Seoul National University, Korea. From 2001 to 2004, he was a Research Staff Member with IBM T.J. Watson Research Center, Yorktown Heights, NY, USA. He joined the School of Electrical Engineering, KAIST, Korea, in 2004, where he is currently a Professor. He was a General Chair of ASP-DAC 2018, has served as Program Chair of ICCD 2014 and VLSI-SoC 2015, and has served as organizing committees and technical program committees of many conferences including DAC, ICCAD, ASP-DAC, ISLPED, and so on. He has served as Associate Editor of IEEE Trans. CAD, IEEE Design & Test, and ACM TODAES. He is an IEEE fellow.

(Invited by Editor-in-Chief: *Atsushi Takahashi*)