Journal of
Micro/Nanolithography,
MEMS, and MOEMS

Nanolithography.SPIEDigitalLibrary.org

Laplacian eigenmaps- and Bayesian clustering-based layout pattern sampling and its applications to hotspot detection and optical proximity correction

Tetsuaki Matsunawa
Bei Yu
David Z. Pan

SPIE.

Tetsuaki Matsunawa, Bei Yu, David Z. Pan, "Laplacian eigenmaps- and Bayesian clustering-based layout pattern sampling and its applications to hotspot detection and optical proximity correction," *J. Micro/Nanolith. MEMS MOEMS* **15**(4), 043504 (2016), doi: 10.1117/1.JMM.15.4.043504.

Laplacian eigenmaps- and Bayesian clustering-based layout pattern sampling and its applications to hotspot detection and optical proximity correction

Tetsuaki Matsunawa,[a,*] Bei Yu,[b] and David Z. Pan[c]
[a]Toshiba Corporation, Yokohama 247-8585, Japan
[b]The Chinese University of Hong Kong, Computer Science and Engineering Department, NT, Hong Kong
[c]The University of Texas at Austin, Electrical and Computer Engineering Department, Austin, Texas 78712, United States

Abstract. Effective layout pattern sampling is a fundamental component for lithography process optimization, hotspot detection, and model calibration. Existing pattern sampling algorithms rely on either vector quantization or heuristic approaches. However, it is difficult to manage these methods due to the heavy demands of prior knowledge, such as high-dimensional layout features and manually tuned hypothetical model parameters. We present a self-contained layout pattern sampling framework, where no manual parameter tuning is needed. To handle high dimensionality and diverse layout feature types, we propose a nonlinear dimensionality reduction technique with kernel parameter optimization. Furthermore, we develop a Bayesian model-based clustering, through which automatic sampling is realized without arbitrary setting of model parameters. The effectiveness of our framework is verified through a sampling benchmark suite and two applications: lithography hotspot detection and optical proximity correction. [DOI: 10.1117/1.JMM.15.4.043504]

Keywords: lithography; pattern sampling; clustering; optical proximity correction; hotspot detection.

Paper 16113 received Jul. 4, 2016; accepted for publication Sep. 28, 2016; published online Oct. 21, 2016.

1 Introduction

As the feature size of semiconductor transistors continues to shrink, it is more and more important to verify the complicated mask so that the overall process cost can be reduced and the manufacturing yield can be improved. Machine learning-based techniques have been demonstrated to be effective in several integrated circuit (IC) manufacturing applications,[1] such as mask optimization,[2] hotspot detection,[3] and lithography verification.[4] The common goal of these methods is to learn a highly accurate prediction model with a small amount of data. Apart from the development of learning algorithm, an effective layout pattern sampling method is also critical to these industrial applications, as the types of training or test data will greatly affect the prediction model performance.

To reduce the training time of mask synthesis and process model calibration, a minimum set of test patterns shall be extracted and sampled to reflect key characteristics in real layouts while maintaining high prediction accuracy.[5] For example, in a hotspot detection problem, balanced test patterns between nonhotspots and real hotspots are required to prevent the overfitting issue.[6] However, automatically extracting essential components from real layouts tends to be difficult because there are innumerable pattern variations in real layouts and the number of dimensions in layout data is high. This is known as an unsupervised problem in machine learning in which some hidden structures must be determined from the given unlabeled data.

Figure 1 illustrates a typical flow of layout pattern sampling. Given input layout, first the feature extraction transfers the geometric information into a set of high-dimensional vectors. Then, dimension reduction is to identify the critical features. Finally, on the simplified dimension space, clustering is carried out to select the sampling results.

So far, several pattern sampling works have been proposed to acquire a set of test patterns. Some clustering techniques have been proposed for test pattern sampling.[7–9] These related works contribute to design automation by extracting feature vectors that represent characteristics of layout patterns and training a classification model based on the feature vectors. However, it is difficult to directly apply an identical clustering technique to different sampling problems, with the following two reasons. First, a criterion for defining pattern similarity to evaluate essential characteristics in real layouts is unclear. Second, most clustering algorithms require several preliminary experiments because there are some parameters that must be tuned in advance, such as the total number of clusters.

In this paper, we propose a pattern sampling framework for creating appropriate test patterns from a given layout effectively. Our key contributions are summarized as follows:

- We develop an efficient feature comparison method with nonlinear dimensionality reduction technique with kernel parameter optimization.
- We develop an automated pattern sampling method using Bayesian model (BM)-based clustering without manual parameter tuning.
- We demonstrate promising test pattern extraction under industrial-strength test chips.

*Address all correspondence to: Tetsuaki Matsunawa, E-mail: tetsuaki.matsunawa@toshiba.co.jp

1932-5150/2016/$25.00 © 2016 SPIE

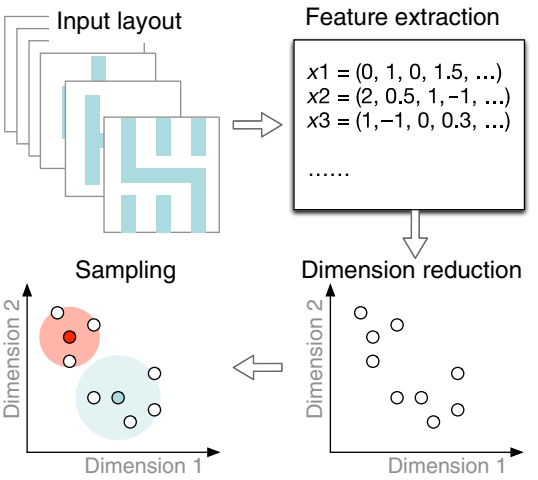

Fig. 1 Example of layout pattern sampling.

The rest of the paper is organized as follows. Section 2 gives the problem formulation. Section 3 introduces the overall flow of layout pattern sampling. Sections 4 and 5 present details of two key algorithms, dimensionality reduction and clustering. Section 6 lists the experimental results, followed by the conclusion in Sec. 7.

2 Problem Formulation

To quantify the sampling performance and to compare diverse layout feature types, a clustering result evaluation method is needed. In this work, we apply Bayes error (BE)[10] to evaluate the degree of overlapping clusters based on Bayes' theorem. BE is defined as follows:

$$\mathrm{BE} = \int \min\{1 - P(\omega_k|\mathbf{x})\}P(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is a given feature vector, $P(\omega_k|\mathbf{x})$ is a conditional probability in class $\omega_k(i = k, \ldots, K)$ that indicates a probability of erroneously determining $\mathbf{x}$, K is the total number of classes, and $P(\mathbf{x})$ is a prior probability of $\mathbf{x}$. BE accurately expresses a quality of distributions among clusters.

The problem formulation of layout pattern sampling is given as follows.

Problem 1 (Layout pattern sampling) Given a layout data, a classification model is trained to extract representative patterns. The goal is to classify the layout patterns into a set of classes minimizing the BE.

Layout pattern sampling can be realized by vector quantization, which maps data sets of vector representations to a limited number of representative patterns called "centroids." The main algorithm to acquire the representative patterns is clustering, which is an unsupervised learning toward a classification model that sorts given data into multiple categories. It should be noted that it is easy to recognize one-dimensional layout patterns, but for general two-dimensional random layout patterns, sampling is very difficult.

3 Overall Flow

The overall flow of our automated layout pattern sampling is illustrated in Fig. 2, which consists of "sampling" phase and "application" phase. In the sampling phase, first a set of features are extracted from the given layout. Then, dimensionality reduction is carried out to simplify the feature space (Sec. 4). Finally, all the features are clustered, and the patterns located in the center of each cluster are used as test patterns (Sec. 5). In the application phase, extracted test patterns are used for various purposes, such as prediction model training for lithography hotspot detection, mask optimization or process simulation, and so on. Note that the quality of extracted patterns can be measured through several applications on layout level, such as hotspot detection, mask optimization, and wafer inspection.

In the layout feature extraction, different from conventional window-based scanning, a design rule check (DRC)-based feature point generation is proposed to identify the key windows. Therefore, we can reduce the scanning window number. In addition, our framework is robust enough that all the feature extraction techniques in previous works (e.g., Ref. 3) can be seamlessly integrated. Feature extraction is one of the most important factors in machine learning applications because the prediction model performance is mostly determined by the types of layout features. It is, however, difficult to define an appropriate layout feature in advance since the optimal characteristics for layout representation vary in different applications. For this reason, feature comparison is important to find out proper method of representing layout patterns by evaluating different types of features.

3.1 *Feature Point Generation*

To extract layout feature efficiently, we propose a DRC-based feature point generation method. Here, we briefly describe the method to locate all feature points for wiring layout. First, all polygons are parsed from a given layout, and then the corners of the polygons are recognized. Next, all polygons are divided into rectangles according to the corners. Then, feature points are generated in both the center of

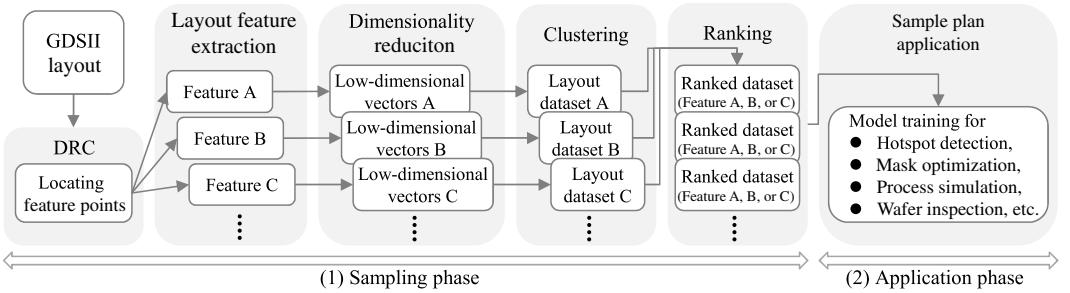

Fig. 2 The overall CAD flow for pattern sampling.

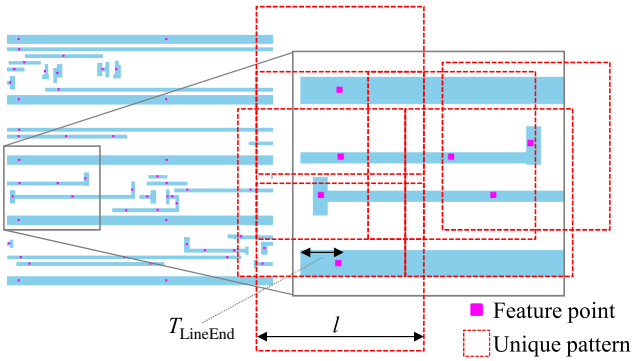

Fig. 3 Determine all feature points.

all rectangles and the positions that are a T_{LineEnd} distance apart from the middle of the short sides of all rectangles. All rectangles, whose long sides exceed a certain threshold T_{LongEdge}, are further divided into two parts evenly. Finally, a set of unique feature points are obtained by filtering duplicated patterns using exact pattern grouping of the patterns within a radius l in the feature points. Figure 3 shows an example of feature points. Although we set the DRC parameters T_{LineEnd}, T_{LongEdge}, and l to 200, 1200, and 1000 nm, respectively, these parameters are freely set when the feature points cover all possible combinations of the layout patterns.

Note that if there is prior knowledge of the unique feature points, the feature point generation phase can be skipped.

3.2 *Feature Extraction*

In this paper, we use the three types of layout features.

3.2.1 *Density-based feature extraction*

The density-based feature represents pattern information based on an area density. Feature vectors $\mathbf{x}$ show arrangement of area values of layout patterns in a given grid as shown in Fig. 4(b). This feature has been used successfully for machine learning-based hotspot detection problems.[11] Parameters of a feature consist of the total size of the encoding area l and the number of grids g. The total dimensions of the feature vectors d in Fig. 4(b) are given by $25(d = g^2)$. Compared with the other features, there is a possibility that generalization capability of a prediction model deteriorates because the geometrical information of layout patterns is locally averaged as the area value. In contrast, the feature has an advantage in that the amount of data is less because d is relatively small.

3.2.2 *Diffraction-based feature extraction*

The diffraction order distribution represents pattern information based on a Fourier spectrum. Feature vectors $\mathbf{x}$ show

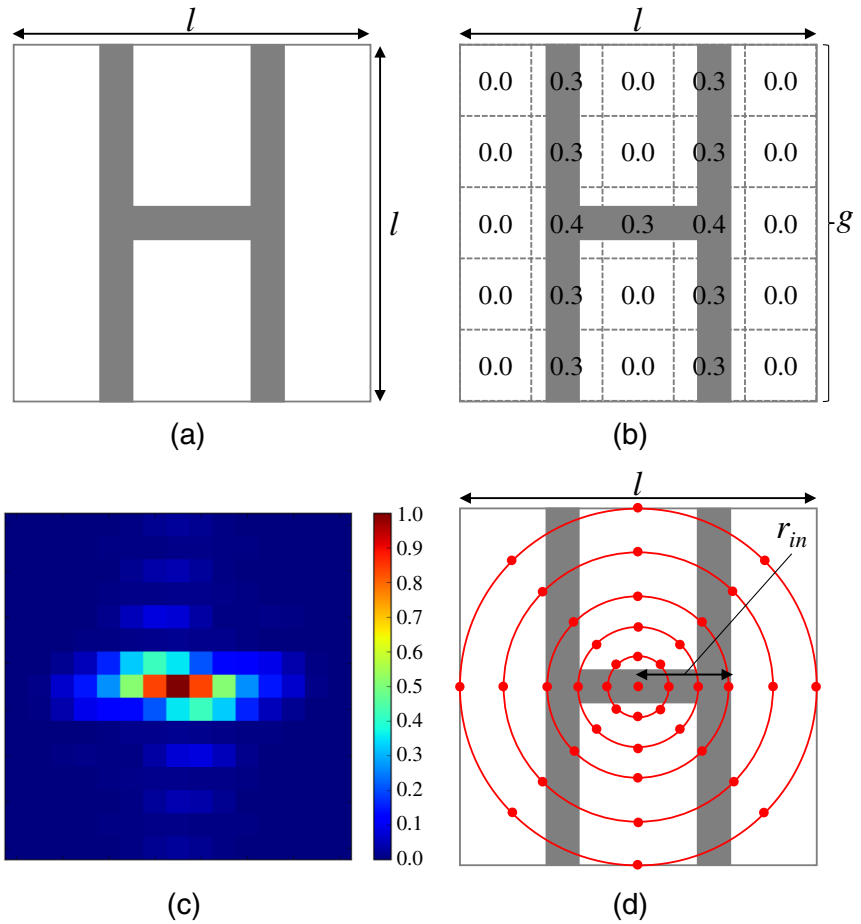

Fig. 4 Layout features: (a) test pattern, (b) density-based feature, (c) diffraction-based feature, and (d) CCS.

arrangement of the coefficients of the Fourier transforms of layout patterns. This feature is widely used for the lithography process optimization.[5] Parameters of a feature consist of the size of the encoding area l, the wavelength λ, and the numerical aperture (NA) in projection optics. Figure 4(c) indicates the diffraction image of the "H"-shaped test pattern. In Fig. 4(c), $\mathbf{x}$ are normalized and consist of the Fourier spectrum in the pupil defined by $(\mathrm{NA}/\lambda)^2$. Thus, d is given by $(2l \times \mathrm{NA}/\lambda)^2$ and in Fig. 4(c), $d = 225$, where $\mathrm{NA} = 1.35$, $\lambda = 193\,\mathrm{nm}$, and $l = 1000\,\mathrm{nm}$. It can be expected to achieve a highly accurate prediction model for lithography process optimization problems because the feature includes an effect from the projection optics. However, prediction model training might be difficult owing to the tendency of dimensions to increase.

3.2.3 *Concentric circle sampling*

This feature corresponds to pattern information that affects propagation of diffracted light from a mask pattern.[12] Feature vectors $\mathbf{x}$ contain subsampled pixel values on concentric circles of layout patterns. Figure 4(d) indicates the basic concept of the CCS of "H"-shaped test pattern. Parameters of a feature consist of the total size of the encoding area l and the sampling density controlling parameter r_{in}. The radius of the concentric circles is $0, 2, 4, \ldots, r_{\mathrm{in}}, r_{\mathrm{in}} + 4, r_{\mathrm{in}} + 8, \ldots, l/2$ pixels, respectively. In this paper, r_{in} is set to 60 to selectively sample the pixel values within the range, in which one order diffraction light is influenced. Thus, the total number of dimensions in Fig. 4(d) is 369, where $l = 1000\,\mathrm{nm}$. It can be expected to achieve a high generalization capability because the feature can correctly express a positional relationship to layout patterns. Also, the subsampled pixel values correspond to important physical phenomena, because diffracted light from a mask pattern is propagated concentrically. However, prediction model training might be difficult because of high-dimensional feature space.

4 Dimensionality Reduction

The total number of dimensions in a layout feature can be more than thousands for some complexed feature types (e.g., image feature). In such high-dimensional space, it is extremely difficult to train a prediction model due to the concentration on the sphere issue.[13] As the dimensions increase, the data are approximately concentrated on the surface of the hypersphere. Because the distance between two data points will be equivalent to the distance between other data points, distinguishing data in high-dimensional space are thereby difficult. Principal component analysis (PCA), which reduces dimensions by transforming data into values of a linearly uncorrelated axis,[6] is the most commonly used dimensionality reduction technique. Although PCA allows us to reduce high-dimensional feature vectors into a lower-dimensional space, it has a disadvantage in that the existing cluster structure in original data is not preserved. We will further discuss the disadvantage of linear dimensionality reduction in Sec. 4.2. To avoid this issue and to handle different types of layout features, we propose an effective nonlinear dimensionality reduction technique.

4.1 *Laplacian Eigenmaps*

Our nonlinear dimensionality reduction technique is based on a Laplacian eigenmaps (LE)[14] while preserving the existing cluster structure. LE effectively reduces complicated feature structures using a kernel method. The embedded matrix $\Psi = (\psi_{n-1}, \psi_{n-2}, \ldots, \psi_{n-m})^{\mathrm{T}}$ is calculated by solving the following generalized eigenvalue problem:

$$L\psi = \gamma D\psi, \tag{2}$$

where $L = D - W$ is the Laplacian matrix, $D = \mathrm{diag}(\sum_{i'=1}^{n} W_{i,i'})$ is the diagonal matrix, γ is the matrix of the eigenvalues in $(\gamma_1 \geq \ldots \geq \gamma_n)$, and $W_{i,i'}$ is the kernel representing a similarity matrix for k-nearest neighbors defined as 1 if $x_i \in \mathrm{kNN}(x_{i'})$ and 0 otherwise. Compared with PCA, LE can effectively map original data into a lower-dimensional space while maintaining the existing cluster structure. Furthermore, it is advantageous in that it can be applied to any kind of feature vectors because the kernel design provides a lot of flexibility. In contrast, because LE uses a kernel method, characteristics of embedded feature space highly depend on the kernel parameter setting. In this paper, we propose an automatic kernel parameter optimization method based on the difference between input feature vectors and an embedded feature vectors.

4.2 *Kernel Parameter Optimization*

Density ratio estimation is a method to directly estimate the density ratio between the two probability distributions without each probability distribution. The kernel parameter can be optimized through the density ratio estimation. We optimize the kernel parameter with the Kullback–Leibler importance estimation procedure (KLIEP) because the optimization problem involved in KLIEP is convex.[15] The density ratio of the probability distribution $P(\mathbf{x})$ and $P'(\mathbf{x})$ of data $\mathbf{x}$ is defined as $r(\mathbf{x}) = P'(\mathbf{x})/P(\mathbf{x})$. In KLIEP, the estimated ratio $\hat{r}$ is defined as the following linear model: $\hat{r}(\mathbf{x}) = \sum_j^b w_j \phi_j(\mathbf{x})$, where w is the parameter to be learned from data samples, $\phi_j(\mathbf{x})$ is the similarity kernel, and b is the total number of data. The parameter w is determined so as to minimize the Kullback–Leibler (KL) divergence described as $\mathrm{KL}[P'(\mathbf{x}) \| \hat{P}'(\mathbf{x})]$. The minimization of the KL divergence is equivalent to maximizing the following: $\int P'(\mathbf{x}) \log(\hat{r}(\mathbf{x}))\mathrm{d}\mathbf{x}$, under the following constraint: $\int \hat{r}(\mathbf{x}) P(\mathbf{x})\mathrm{d}\mathbf{x} = 1$. By approximating the expectation with sample average, the following convex optimization problem is derived:

$$\max_{w} \sum_{i=1}^{n'} \log[w^T \phi(\mathbf{x}'_i)], \tag{3}$$

$$\text{subject to} \sum_{i=1}^{n} w^T \phi(\mathbf{x}_i) = n \quad \text{and} \quad w \geq 0, \tag{4}$$

where n' and n are the test input samples and the training input samples in a likelihood cross-validation,[16] respectively. Then, we can obtain the unique global solution by simply performing gradient ascent and feasibility satisfaction iteratively.[15] Meanwhile, the kernel parameter can be learned using the likelihood cross-validation method by

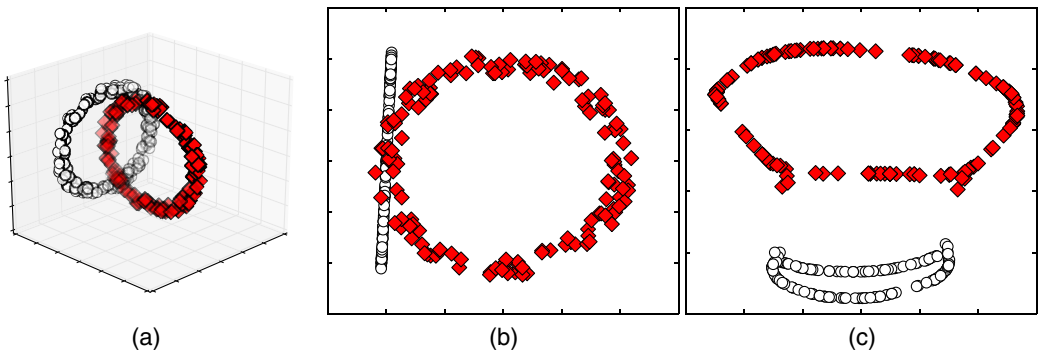

Fig. 5 Comparison of dimensionality reduction: (a) test feature, (b) PCA result, and (c) LE result.

approximating the unbiased estimator of the KL divergence. In this paper, the kernel parameter, the number of k-nearest samples in the similarity matrix W, is optimized by using the given feature vectors as $P(\mathbf{x})$, and the embedded feature vectors as $P'(\mathbf{x})$.

Figure 5 shows the difference between linear and nonlinear dimensionality reduction methods. The red data in Fig. 5(a) indicate a ring-shaped test feature and the gray shows a ring-shaped test feature intersecting the red data in three-dimensional space. Figures 5(b) and 5(c) indicate dimensionally reduced data by PCA and dimensionally reduced data by LE, respectively. The figures show that if a data include complicated nonlinear cluster structures, there is a lose in linear dimensionality reduction technique. In our proposed framework, by combining LE and KLIEP, dimensionally reduced feature data can be obtained without arbitrary parameter tuning, while also preserving the existing cluster structure.

5 Bayesian Clustering

As mentioned in the introduction, the need for a method of determining the total number of clusters K continues one of the major issues concerning conventional clustering algorithms. Although several K estimation methods have been proposed, it is difficult to manage these methods. For example, the Jain–Dubes method is proposed[17] for K estimation in K-means clustering, which is a well-known and widely used clustering algorithm. However, this method does not work well if the feature space is complicated and consists of nonlinearly distributed clusters, because K-means is known to be a local-minimum solution and assumes that each cluster is a hypersphere.

To overcome the above issues, we propose a BM-based clustering method. For the clustering task, there are many unknown parameters, such as the number of clusters, the cluster labels, the cluster shapes, and the cluster parameters including a mean or a variance. In a BM approach, all unknown parameters can be naturally learned from a given data by expressing a parameter distribution as an infinite dimensional discrete distribution. Specifically, we first consider an infinite Gaussian mixture model in which data $\mathbf{x}$ is generated

$$P(\mathbf{x}|\alpha, P(\theta)) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}(\mu_k, \sigma_k^2), \tag{5}$$

where $P(\theta)$ is the prior distribution of parameters θ, α is the learnable hyperparameter, and π is the mixing ratio. Note that θ includes the parameters of Eq. (5), such as the parameters of Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k^2)$ with the mean μ_k and the variance σ_k^2 and the mixing ratio π_k, where $\sum_{k=1}^{\infty} \pi_k = 1$. The BM considers that all data are automatically classified while generating each data $\mathbf{x}$ from any of infinite mixture distributions. When a cluster label z_n of data $\mathbf{x}_n$ is unknown, the posterior probability of z_n is given based on Bayes' theorem

$$P(z_n = k|\mathbf{x}_n, z_1, \ldots, z_{n-1}) \propto P(\mathbf{x}_n|z_n) P(z_n|z_1, \ldots, z_{n-1}). \tag{6}$$

This equation can be written in the form

$$P(z_n = k|\mathbf{x}_n, z_1, \ldots, z_{n-1}) \propto \begin{cases} P(\mathbf{x}_n|k) \frac{n_k}{\alpha+n-1} & (k = 1 \cdots K), \\ P(\mathbf{x}_n|k_{\text{new}}) \frac{\alpha}{\alpha+n-1} & (k = K+1), \end{cases} \tag{7}$$

where n_k is the number of times class k appears within $z_{1:n-1} = (z_1, \ldots, z_{n-1})$, K is the number of current clusters, the first term is the likelihood of x_n and the second term is the prior probability of z_n. This procedure is known as a Chinese restaurant process (CRP), which is a distribution on partitions obtained by imagining a process in which customers share tables in a Chinese restaurant.[18] Specifically, when we consider a restaurant with infinitely many tables, if there are already many people sitting there, a customer is likely to sit at a table with probability proportional to n_k. In contrast, the customer will sit at a new table with probability proportional to α, indicating that a new class k_{new} is generated proportional to α. Note that as $P(z_n|z_1, \ldots, z_{n-1})$ represents the probability of selecting cluster label z_n given $z_{1:n-1}$ while considering the variance of cluster size based on α, the denominator of the prior in Eq. (7) can be written by $\alpha + n - 1$.

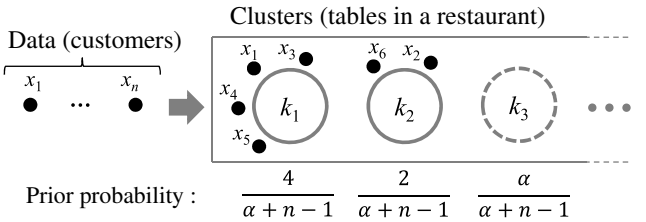

Fig. 6 Overview of the CRP-based cluster selection.

Algorithm 1 Automatic clustering with Gibbs sampling.

Require: X, θ

while not converged **do**

  for n in random permutation $(1, \ldots, N)$ **do**

    Remove $\mathbf{x}_n$ from cluster z_n and update θ;

    Sample $z_n \sim P(z_n|X, Z_{-n})$;

    Add $\mathbf{x}_n$ into cluster z_n and update θ;

  end for

end while

return $z_1, \ldots, z_N$;

Figure 6 shows an overview of the CRP-based cluster selection. By using Eq. (7), when a feature vectors $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are given, a cluster labels of the features $Z = (z_1, \ldots, z_N) : z_n \in 1 \cdots K$ can be solved by using Gibbs sampling,[19] as shown in Algorithm 1. The cluster label Z followed by a true distribution $P(Z|X)$ is given by iteratively sampling the hidden variable z_n from a conditional probability $P(z_n|X, Z_{-n})$, where Z_{-n} is Z without z_n. Note that this is an exchangeable process in that the probability does not depend on the order in x_n.

Though the posterior distribution of the cluster assumes Gaussian distribution, this assumption works as the cluster distribution because the feature vectors are partially preclassified by our nonlinear dimensionality reduction technique. In addition, the hyperparameter α can also be determined automatically by k-fold cross-validation.[16] Furthermore, the proposed framework allows us to easily quantify the clustering results because unknown parameters such as the mean or variance of each cluster can be learned directly from given data. Therefore, in our framework, automatic clustering can be expected without manual parameter tuning.

6 Experimental Results

6.1 *Experimental Setup*

The proposed methodologies were implemented in C++ and Python on a Linux machine with eight 3.4-GHz CPUs and 32-GB memory. `Calibre`[20] was used to perform lithography simulation with wavelength $\lambda = 193$ nm and NA $=$ 1.35. Two 32-nm node industrial layouts, A and B, were applied as benchmark. The areas of the layout A and layout B are 10,092.2 and 12,702.3 μm^2, respectively.

In layout feature extraction, layout feature is important as it determines how to encode initial geometrical information. We implemented three layout features introduced in Sec. 3.2: density-based feature, diffraction-based feature and CCS feature. In density-based feature, for each layout region, we split them into a set of grids and then the densities in the grids are encoded in a vector.[11] In our implementation, the area of layout region is set to 1000 nm, and the grid number in each layout region is set to 10. Diffraction-based feature represents pattern information based on a Fourier spectrum.[5] CCS feature indicates pattern information that affects propagation of diffracted light from a mask pattern. The total feature dimension numbers are 100 for density-based feature, 225 for diffraction-based feature, and 369 for CCS.

In addition to the proposed LE-based dimensionality reduction (Sec. 4), we also implemented a conventional dimensionality reduction technique, PCA. PCA has been applied in several layout analysis works (e.g., Ref. 6).

In the implementation of Bayesian clustering (Sec. 5), the hyperparameter α is determined through fivefold cross-validation, which involves multiple training runs to reduce variability of estimation. In the cross-validation, the data are partitioned into five groups, and then $4 = 5 - 1$ of the groups are used to train a set of models. The remaining group is used to evaluate the models. Then, this procedure is repeated for all five possible choices. The parameter is finally estimated as an average of the five runs. A Gaussian–Wishart distribution is used as the prior distribution because the mean and the variance in each cluster are unknown. Parameters of the priors, prior mean, and prior covariance are set to 0, the covariance of input feature vectors, respectively. The other parameter of the prior, the freedom of Wishart distribution, is also determined with fivefold likelihood cross-validation. The total iterations of the Gibbs sampling are set to 1000 and burn-in is half of the total iterations. We also implemented K-means algorithm, which is a classical clustering method. In the K-means algorithm, the K value is determined by the Jain–Dubes method.[17]

In this paper, we use BE in Eq. (1) to evaluate clustering performance, and here we demonstrate its effectiveness. Note that a well-known method for measuring the clustering performance is the scattering ratio between within-class scatter and between-class scatter (WCS/BCS). Typically, if the resulting WCS is small and the resulting BCS is big, the clusters are indicated to be well organized. However, WCS/BCS lacks the elements necessary for rigorous evaluation because the degree to which clusters overlap cannot be measured.[16] Even if WCS/BCS is small, data separation becomes difficult when there are overlapped cluster distributions. Figure 7 shows examples of four clusters that follow different normal Gaussian distributions. As shown in Fig. 7(a), if the clusters are clearly separated, both BE and WCS/BCS show nearly equal to 0. In contrast, as shown in Fig. 7(b), it is difficult to evaluate the degree of separability by using WCS/BCS when there are overlapped clusters. It should be noted that it is difficult to measure clustering performance using BE in high-dimensional space due to the concentration on the sphere issue[13] (see Sec. 4 in detail). Although quantifying the effective dimensions is difficult, it is shown that BE can successfully estimate error rates for 9-dimensional data, but it fails for 120-dimensional data.[10] However, in our framework, BE is applicable to most layout features including high-dimensional data by combining dimensionality reduction techniques.

6.2 *Effectiveness of Pattern Sampling*

In the first experiment, we verify the effectiveness of the proposed LE and Bayesian clustering. Layout A is used as test the layout to extract layout patterns. Table 1 lists the pattern sampling results with different dimensionality reduction and clustering techniques. "PCA" and "LE" indicate principal component analysis and Laplacian eigenmaps, respectively.

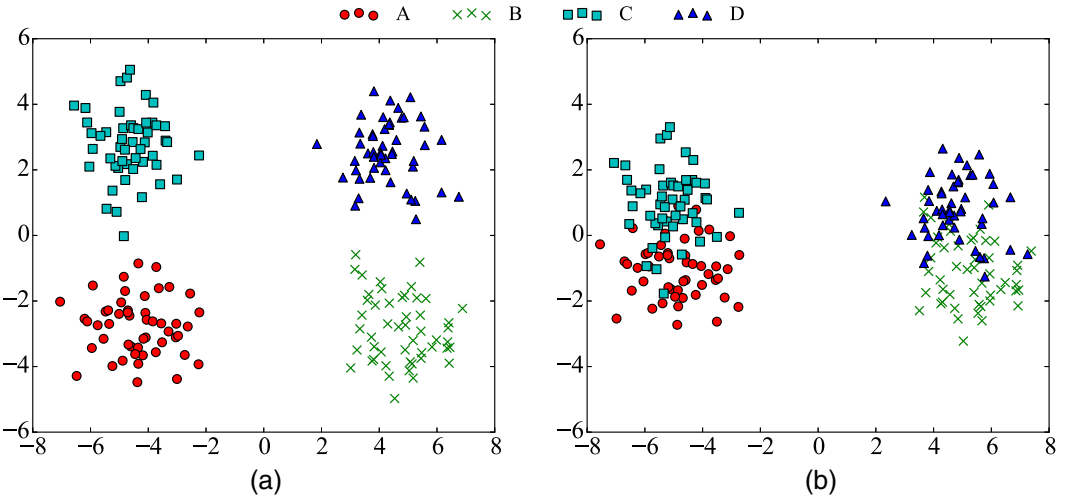

Fig. 7 Comparison between BE and WCS/BCS: (a) case 1: BE = 0.02, WCS/BCS = 0.07 and (b) case 2: BE = 1.68, WCS/BCS = 0.07.

"Km" and "BM" indicate K-means clustering and Bayesian clustering, respectively. Different combination of dimensionality reduction and clustering is tested. For example, column "PCA+Km" means PCA is applied for dimensionality reduction, while K-means is used for clustering. For each combination, columns "K," "BE," and "CPU(s)" give the number of final clusters, the BE defined in Eq. (1), and the runtime in seconds.

From Table 1, we can see that our proposed method ("LE +BM") can achieve the best clusters in terms of BE. Though the combination with LE and K-means is known as a spectral-clustering, defining K remains difficult. It can also be seen that K in our method tends to be slightly higher than K-means-based methods. It should be noted that there is no correct number of clusters because pattern sampling is an unsupervised learning task. We apply obtained samples (test patterns) to the next two applications to evaluate the effectiveness of the sampling framework. It should be also noted although BM is time-consuming, the runtime can be reduced by using variational Bayes inference,[13] which is a subject for future work. Figure 8 shows a general view of layout A and several examples of sampled representative patterns. The gray features are design pattern, and the red dots indicate representative patterns obtained by our sampling techniques. From Fig. 8, we can see the sampling results different from each other in geometrical shape.

6.3 *Effectiveness on Hotspot Detection Application*

In the second experiment, we verify our sampling methodologies in hotspot detection application, where hotspots need to be quickly detected without lithography simulation. Applied in early physical design stage, hotspot detection[3,11,21] can effectively reduce the turn-around time (TAT) and the design cost. A hotspot detection model was trained with the test patterns in the layout A obtained in the sampling experiments and the hotspots that are detected by our industry setting verification process. Then, the layout B was scanned using the detection model.

Although many effective algorithms have been proposed, such as artificial neural network and support vector machine,[3,11] we focus on a specific detection algorithm to evaluate the effect of the samples obtained by our sampling framework since proposal of the optimal detection algorithm for the hotspot detection problem is not the intent of this paper. Furthermore, other algorithms proposed by Ding et al.[3] and Lin et al.[11] are not compared in this paper. The reason of such limitation is that it is difficult to measure

Table 1 Comparison of pattern sampling techniques.

	PCA+Km			LE+Km			PCA+BM			LE+BM		
Layout B	K	BE	CPU(s)	K	BE	CPU(s)	K	BE	CPU(s)	K	BE	CPU(s)
Density	4	143.4	0.4	5	1198.0	99.8	8	82.7	29.9	11	57.7	130.5
Diffraction	4	230.7	0.7	6	898.3	100.8	13	183.9	39.5	19	117.9	148.0
CCS	8	125.8	1.0	13	4.0E + 06	307.4	10	162.3	30.9	13	70.8	345.3
Average	5	166.6	0.7	8	1.3E + 06	169.3	10	143.0	33.4	14	82.1	207.9
Ratio	—	**1.0**	—	—	≫ **1.0**	—	—	**0.9**	—	—	**0.5**	—

Note: Bold values emphasize the efficacy of our method.

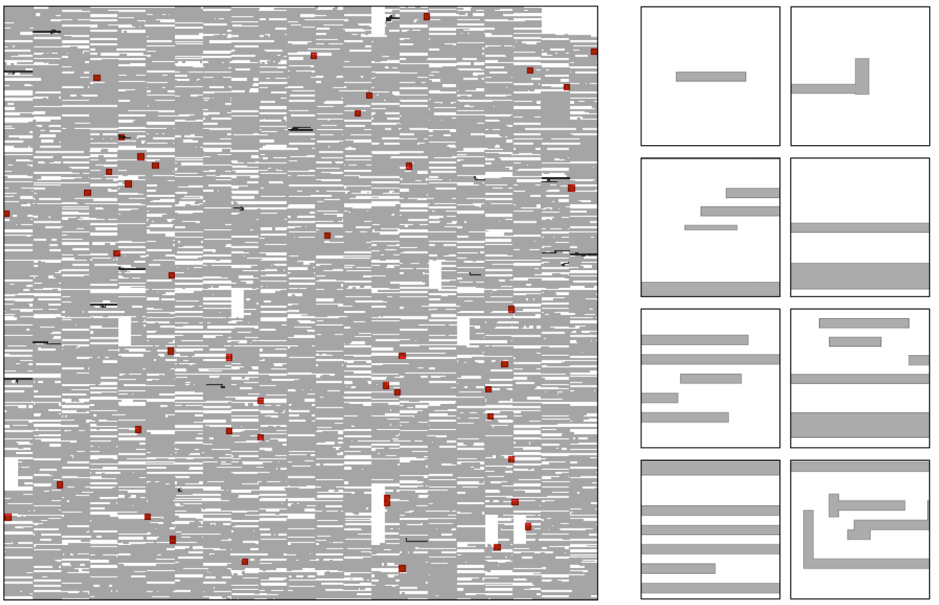

Fig. 8 Sampling results: (a) layout A and (b) examples of sampled representative patterns.

the performance of hotspot detection for actual full-chip layout because the public benchmarks used in the related works consist only of limited clipped layouts.[22] This paper uses the AdaBoost classifier, which has shown relatively better performance compared to other classifiers.[4,23]

Figure 9 gives the hotspot detection results with different dimensionality reduction and clustering techniques. Two important metrics are used to evaluate the performance of hotspot detection. The first one is the hotspot detection "accuracy" defined as Hit/(no. of hotspots), where Hit is the number of correctly detected hotspots. Another one is the H/E ratio (false alarm) defined as Hit/Extra, where Extra is the number of falsely detected hotspots. From Fig. 9, we can see that the prediction model performance tends to deteriorate according to the increase in BE in terms of false alarm. Moreover, from the point of view of both accuracy and false alarm, the diffraction-based feature is fit to the hotspot detection problem. The results also show that the representative training patterns can be obtained by our framework because the results with our proposed method show the highest accuracy and the lowest false alarm.

6.4 *Effectiveness on Optical Proximity Correction Application*

In the last experiment, we verify our proposed sampling techniques in optical proximity correction (OPC) regression application. OPC is a mask optimization technique to improve image pattern fidelity on a wafer. The most widely used method is model-based OPC in which the displacement amount of fragment movement of a mask pattern is computed based on lithography simulation. Although this method is expected to achieve very high accuracy, it is also known to be extremely time-consuming. To reduce the TAT, linear regression-based OPC is proposed and this showed that it is possible to reduce the iterations in model-based OPC.[24] However, the question of what kind of training patterns should be used for the regression model remains open. To evaluate the performance of OPC regression, we use the root-mean-square prediction error (RMSPE) defined as follows:

$$\mathrm{RMSPE} = \sqrt{(1/N)\sum (y_i - \hat{y})^2}, \tag{8}$$

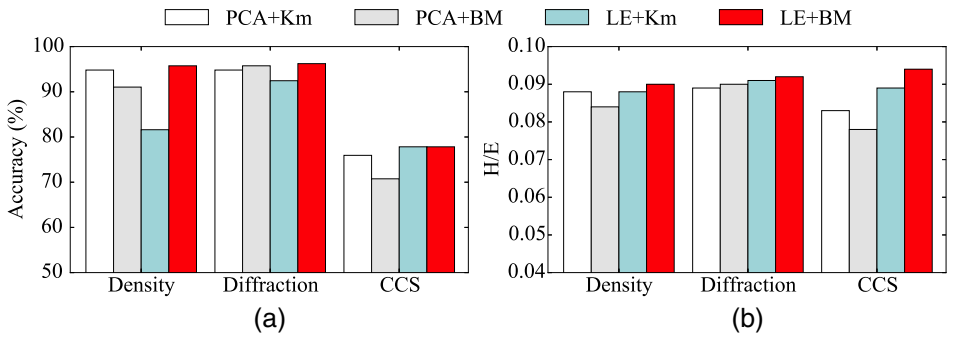

Fig. 9 Hotspot detection result comparison on different techniques: (a) the impact on accuracy and (b) the impact on false alarm.

Table 2 OPC performance comparison.

	PCA+Km		LE+Km		PCA+BM		LE+BM	
Layout A	TD#	RMSPE	TD#	RMSPE	TD#	RMSPE	TD#	RMSPE
Density	308	7.8	282	8.4	526	6.7	577	6.6
Diffraction	211	12.2	508	13.3	1048	11.1	1102	9.7
CCS	573	5.3	461	5.4	687	4.8	668	4.5
Average	364.0	8.4	417.0	9.0	753.7	7.5	782.3	6.9
Ratio	—	**1.0**	—	**1.1**	—	**0.9**	—	**0.8**

Note: Bold values emphasize the efficacy of our method.

where N is the total number of samples, y_i is the fragment movement determined by model-based OPC, and $\hat{y}$ is the predicted fragment movement. Note that although the use of RMSPE is not universal to evaluate OPC performance, it is capable of effectively measuring performance of a linear regression model.

All displacement amounts of the fragments in the layout A and B are computed by the lithography simulation with our industry setting model-based OPC. Then, the displacements of the patterns obtained in the sampling experiments are used to train a linear regression model defined by

$$y = \sum_{i=0}^{b} w_i \mathbf{x} + \epsilon, \tag{9}$$

where w is the coefficient, $\mathbf{x}$ is the feature vector, b is the dimensions, and ϵ is a random factor. Finally, all displacements of the patterns in the layout B are predicted using the regression model. Note that the primary objective of this paper is to sample an appropriate training pattern as an input of regression model. All the initial patterns are sampled through different layout pattern sampling techniques.

Table 2 compares the OPC performances under different sampling techniques. Similar to Table 1, we enumerate different combinations of dimensionality reduction and clustering techniques. For each combination, columns "TD#" and "RMSPE" give the number of training data and RMSPE value through Eq. (8), respectively. Note that the total number of fragments in the layout A, i.e., the TD# before our sampling technique is applied, is 33,127. We can see that the model trained with our test patterns achieved the best prediction accuracy. From Table 2, we can also see that the number of training data can be dramatically reduced by using our method while maintaining high prediction accuracy. Although the "LE+Km" method achieves significant reduction ratio (33,127/417), the corresponding RMSPE value is the worst. It should be noted that compared with the result of the "PCA+BM" method, our method achieved a better RMSPE even though the number of training data is very similar to the case of "PCA+BM." This indicates that the test patterns obtained by our method include sufficient characteristics of whole chip layout even for different types of layout features.

7 Conclusion

In this paper, we proposed a layout pattern sampling framework for IC manufacturing design. By applying our nonlinear dimensionality reduction technique with kernel parameter optimization, dimensionality- and type-independent layout feature can be used in accordance with applications. The BM-based clustering technique is able to classify layout data without manual parameter tuning. The experimental results show that our proposed method can effectively identify the key layout patterns that represent characteristics of whole chip, thus it promises to dramatically reduce both the manufacturing cost and the cost of process optimization. In the future, we expect to extend our framework to more IC manufacturing design applications.

Acknowledgments

This work was supported in part by the National Science Foundation, the Semiconductor Research Corporation, and the Chinese University of Hong Kong Direct Grant for Research.

References

1. B. Yu et al., "Machine learning and pattern matching in physical design," in *IEEE/ACM Asia and South Pacific Design Automation Conf. (ASPDAC'15)*, pp. 286–293 (2015).
2. N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *J. Opt.* **12**(4), 045601 (2010).
3. D. Ding et al., "EPIC: efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *IEEE/ACM Asia and South Pacific Design Automation Conf. (ASPDAC'12)*, pp. 263–270 (2012).
4. Z. Xiao et al., "Directed self-assembly (DSA) template pattern verification," in *ACM/IEEE Design Automation Conf. (DAC)*, pp. 1–6 (2014).
5. S. Shim, W. Chung, and Y. Shin, "Synthesis of lithography test patterns through topology-oriented pattern extraction and classification," *Proc. SPIE* **9053**, 905305 (2014).
6. B. Yu et al., "Accurate lithography hotspot detection based on principal component analysis-support vector machine classifier with hierarchical data clustering," *J. Micro/Nanolith. MEMS MOEMS* **14**(1), 011003 (2015).
7. J. Oberschmidt et al., "Automation of sample plan creation for process model calibration," *Proc. SPIE* **7640**, 76401G (2010).
8. W. C. Tam, O. Poku, and R. D. Blanton, "Systematic defect identification through layout snippet clustering," in *IEEE Int. Test Conf. (ITC'10)*, pp. 1–10 (2010).
9. W. Zhang et al., "Automatic clustering of wafer spatial signatures," in *ACM/IEEE Design Automation Conf. (DAC'13)*, p. 71 (2013).
10. K. Tumer and J. Ghosh, "Estimating the Bayes error rate through classifier combining," in *IEEE Int. Conf. on Pattern Recognition (ICPR'96)*, Vol. **2**, pp. 695–699 (1996).
11. S.-Y. Lin et al., "A novel fuzzy matching model for lithography hotspot detection," in *ACM/IEEE Design Automation Conf. (DAC'13)*, p. 68 (2013).
12. T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical Bayes model," *Proc. SPIE* **9426**, 94260X (2015).
13. C. M. Bishop et al., *Pattern Recognition and Machine Learning*, Vol. **4**, Springer, New York (2006).

14. M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Conf. on Neural Information Processing Systems (NIPS'01)*, Vol. **14**, pp. 585–591 (2001).
15. M. Sugiyama et al., "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Conf. on Neural Information Processing Systems (NIPS'07)*, pp. 1433–1440 (2007).
16. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New Jersey (2012).
17. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., New Jersey (1988).
18. D. Blackwell and J. B. MacQueen, "Ferguson distributions via Pólya urn schemes," *Ann. Stat.* **1**, 353–355 (1973).
19. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**(6), 721–741 (1984).
20. Mentor Graphics, *Calibre Verification User's Manual* (2008).
21. J.-Y. Wuu et al., "Rapid layout pattern classification," in *IEEE/ACM Asia and South Pacific Design Automation Conf. (ASPDAC'11)*, pp. 781–786 (2011).
22. A. J. Torres, "ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD'12)*, pp. 349–350 (2012).
23. T. Matsunawa et al., "A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction," *Proc. SPIE* **9427**, 94270S (2015).
24. A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Trans. Semicond. Manuf.* **21**(2), 263–271 (2008).

Tetsuaki Matsunawa received his PhD in computer science from the University of Tsukuba in 2008. He joined Toshiba Corporation in 2008 and has been working in the area of optical lithography. He visited the University of Texas at Austin as a visiting scholar from 2013 to 2015. His current research interests include design for manufacturability and machine learning algorithms with applications in computational lithography.

Bei Yu is an assistant professor at the Department of Computer Science and Engineering, the Chinese University of Hong Kong. His research interests include design for manufacturability and optimization algorithms with applications in CAD. He received the European Design and Automation Association (EDAA) outstanding dissertation award in 2014, SPIE education scholarship in 2013, IBM PhD scholarship in 2012, best paper awards at ICCAD'13 and ASPDAC'12, and three other best paper award nominations.

David Z. Pan is a professor at the Department of Electrical and Computer Engineering, University of Texas at Austin. His research interests include IC design for manufacturability/reliability/security, new frontiers of physical design, and CAD for emerging technologies. He has published over 250 technical papers and is the holder of eight U.S. patents. He has received many awards, including a Semiconductor Research Corporation technical excellence award and 13 best paper awards, among others. He is an IEEE fellow.