Efficient Spatial Variation Modeling via Robust Dictionary Learning

Changhai Liao[1], Jun Tao*[1], Xuan Zeng*[1], Yangfeng Su[2], Dian Zhou[1, 3] and Xin Li[4]
[1] ASIC & System State Key Lab, Dept. of Microelectronics, Fudan University, Shanghai, China,
[2] School of Mathematical Sciences, Fudan University, Shanghai, China,
[3] Dept. of EE, University of Texas at Dallas, Dallas, USA,
[4] Dept. of ECE, Carnegie Mellon University, Pittsburgh, USA

***Abstract*— In this paper, we propose a novel spatial variation modeling method based on robust dictionary learning for nanoscale integrated circuits. This method takes advantage of the historical data to efficiently improve the accuracy of wafer-level spatial variation modeling with extremely low measurement cost. Robust regression is adopted by our implementation to reduce the bias posed by outliers. An iterative coordinate descent method is further introduced to solve the dictionary learning problem with consideration of missing data. Our numerical experiments based on industrial measurement data demonstrate that the proposed method achieves up to 70% error reduction over the conventional VP approach without increasing the measurement cost.**

Keywords—Process variaiton; dictionary training; robust regression

I. Introduction

With the continued scaling of CMOS technology, process variation has become the major roadblock for integrated circuits (ICs) [1]. The increasing fluctuations posed by IC manufacturing process lead to significant performance variations and substantial yield loss [2]. To address this issue, variation modeling and characterization is an important task for today's integrated circuits.

Recently, several statistical methods, such as Virtual Probe (VP) [3]-[10] and Gaussian process (GP) [11], have been developed to model spatial variations with low measurement cost. Taking VP as an example, it exploits the sparse representation in frequency domain to minimize the required measurement data for spatial variation modeling [7]. VP relies on discrete cosine transform (DCT) [20]. It approximates the spatial variations of a wafer by the linear combination of DCT basis functions and assumes that a large number of DCT coefficients are close to zero.

The DCT basis functions used by VP are generic. They can be applied to different wafers without knowing their spatial variations in advance. However, a number of historical wafers are often available in practice. These wafers carry the important knowledge about the manufacturing process that we aim to characterize. The question here is how to take advantage of the historical information to further improve the accuracy and/or reduce the cost for our application of spatial variation modeling.

* Corresponding authors: { taojun, xzeng }@fudan.edu.cn.

Towards this goal, we adopt the dictionary learning technique from the statistics community [12]-[13]. The key idea is to learn a set of specific basis functions (also known as *dictionary*) based on the historical data, instead of relying on the general-purpose DCT basis functions. Since these specific basis functions carry the unique information of a particular manufacturing process, they are expected to model the spatial variations more accurately than the general-purpose DCT basis functions.

The aforementioned dictionary learning problem, however, is not trivial. Most conventional dictionary learning algorithms (e.g., K-SVD [13]) cannot be directly applied here, because silicon measurements are not ideal – they often contain outliers (i.e., measurement points that are distant from the other measurements and have large measurement errors) and missing data. To accommodate these non-idealities, we propose a novel Robust Dictionary Learning method in this paper. In particular, we incorporate the idea of robust regression into dictionary learning so that the resulting dictionary is not highly biased by the outliers, i.e., we apply the robust l_1-norm in the minimization of the modeling error, rather than outlier-sensitive l_2-norm minimization [17]. Furthermore, we mathematically derive a new dictionary learning formulation with consideration of missing data. The proposed dictionary learning problem can be efficiently solved by an iterative coordinate descent method [18] that iteratively solves a sequence of linear programming problems. As will be demonstrated by the industrial examples in section V, our proposed approach achieves up to 70% error reduction over the conventional VP method.

The remainder of this paper is organized as follows. In Section II, we briefly review the background of VP and the existing dictionary learning method. In Section III, we develop a Robust Dictionary Learning method to deal with outliers and missing data for spatial variation modeling. Next, we discuss several implementation issues in Section IV. The efficacy of the proposed method is demonstrated by two industrial examples in Section V. Finally, we conclude in Section VI.

II. Background

A. Virtual Probe

Without loss of generality, in order to intuitively characterize the spatial variation, an interested performance (e. g., the frequency of a ring oscillator) can be expressed as 2-D function $g(x, y)$, where x and y represent the coordinates of

spatial location on a wafer. After discretization, the coordinates x and y can be denoted as integers $x \in \{1, 2, ..., P\}$ and $y \in \{1, 2, ..., Q\}$[7]. To capture the information in spatial frequency domain, $g(x, y)$ can be mapped to frequency domain by several kinds of transforms, such as Fourier Transform, DCT and wavelet transform. In VP [7], DCT transform is taken as an example.

Applying the DCT basis functions [20], it is easy to verify that once all the sampling values $\{g(x, y); x = 1, 2, ..., P, y = 1, 2, ..., Q\}$ are known, the DCT coefficients can be uniquely determined, and vice versa. However, in order to reduce testing cost, VP would like to recover $\{g(x, y); x = 1, 2, ..., P, y = 1, 2, ..., Q\}$ accurately from an limited number of samples at locations $\{(x_m, y_m); m = 1, 2, ..., M\}$, where $M << PQ$. So the recovery can be formulated as a linear equation [7]:

$$\boldsymbol{A}_D \boldsymbol{\eta} = \boldsymbol{B}, \tag{1}$$

where

$$\boldsymbol{A}_D = \begin{bmatrix} A_{1,1,1} & A_{1,1,2} & ... & A_{1,P,Q} \\ A_{2,1,1} & A_{2,1,2} & ... & A_{2,P,Q} \\ \vdots & \vdots & \vdots & \vdots \\ A_{M,1,1} & A_{M,1,2} & ... & A_{M,P,Q} \end{bmatrix}, \tag{2}$$

$$A_{m,u,v} = \alpha_u \cdot \sigma_v \cdot \cos\frac{\pi \cdot (2x_m - 1) \cdot (u-1)}{2 \cdot P} \cdot \cos\frac{\pi \cdot (2y_m - 1) \cdot (v-1)}{2 \cdot Q}, \tag{3}$$

$$\alpha_u = \begin{cases} \sqrt{1/P} & (u = 1) \\ \sqrt{2/P} & (2 \le u \le P) \end{cases}, \ \sigma_v = \begin{cases} \sqrt{1/Q} & (v = 1) \\ \sqrt{2/Q} & (2 \le v \le Q) \end{cases} \tag{4}$$

$$\boldsymbol{\eta} = [G(1,1) \quad G(1,2) \quad ... \quad G(P,Q)]^T, \tag{5}$$

$$\boldsymbol{B} = [g(x_1, y_1) \quad g(x_2, y_2) ... g(x_M, y_M)]^T. \tag{6}$$

Here, matrix $\boldsymbol{A}_D$ denote the DCT basis. If all the unknown DCT coefficients $\boldsymbol{\eta} = \{G(u, v); u = 1, 2, ..., P, v = 1, 2, ..., Q\}$ can be calculated from (1) based on the sampling data at locations $\{(x_m, y_m); m = 1, 2, ..., M\}$, all the function values $\{g(x, y); x = 1, 2, ..., P, y = 1, 2, ..., Q\}$ can be obtained by inverse discrete cosine transform (IDCT) [20]. However, solving (1) is not trivial since $M << PQ$, i.e., the equations in (1) are profoundly underdetermined and cannot be uniquely solved by a simple matrix inverse [7].

Considering the sparse property of spatial variation, it is rational to assume that most of coefficients in $\boldsymbol{\eta}$ are close to zero [7]. Then VP solves (1) via formulating it as an l_1-norm regularization program in compressive sensing field, namely:

$$\begin{aligned} \min \quad & \left\| \boldsymbol{A}_D \boldsymbol{\eta} - \boldsymbol{B} \right\|_2 \\ s.t. \quad & \left\| \boldsymbol{\eta} \right\|_1 \le \lambda_0 \end{aligned}, \tag{7}$$

where $\|\bullet\|_1$ represents the l_1-norm of a vector, i.e., the sum of absolute values of all elements in the vector. $\|\bullet\|_2$ represents the l_2-norm of a vector, i.e., the square root of the sum of squared values for all the elements in the vector. λ_0 represents the sparse level of $\boldsymbol{\eta}$. Once the DCT coefficients $\boldsymbol{\eta}$ are calculated, the spatial variation model can be constructed by IDCT.

B. Dictionary Learning

Compared with the modeling methods using general-purpose basis functions (e.g., DCT basis adopted in VP [19]), the dictionary learning algorithms have been proved to be much more efficient in statistics community [12]-[13]. Suppose that we have N measurement examples $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N]$ and $\boldsymbol{s}_i$ could be sparsely represented with a specific dictionary $\boldsymbol{D}$. Applying the conventional dictionary learning methods, finding the best dictionary $\boldsymbol{D}$ to represent these examples is equivalent to solve:

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{\beta}} \quad & \sum_{i=1}^{N} \left\| \boldsymbol{D}\boldsymbol{\beta}_i - \boldsymbol{s}_i \right\|_2^2 \\ s.t. \quad & \left\| \boldsymbol{\beta}_i \right\|_0 < T_0 \end{aligned}, \tag{8}$$

where $\|\bullet\|_0$ represents the l_0-norm of a vector. So $\|\boldsymbol{\beta}_i\|_0$ means the number of non-zeroes in each coefficient vector $\boldsymbol{\beta}_i$, $i = 1, 2, ..., N$, and T_0 represents its sparse level. Several different dictionary learning methods have been proposed to solve (8), such as the method of optimal directions (MOD) and K-SVD algorithm. Among these methods, K-SVD algorithm has gained popularity in recent years due to its fast convergence rate [13]. Algorithm 1 summarizes its basic flow, where K is the number of basis functions estimated experimentally or determined by using cross validation technique [20].

Algorithm 1: K-SVD Algorithm

1. *Initialization*: Initialize the dictionary matrix $\boldsymbol{D}^0$ with l_2 normalized columns, and set the iteration counter $J = 1$.
2. Repeat until the dictionary matrix $\boldsymbol{D}$ converges:
3. *Sparse coding stage*: Use any sparse regression algorithm to update the coefficient vector $\boldsymbol{\beta}^J{}_i$ at the J-th iteration by solving the optimization problem,

$$\min_{\boldsymbol{\beta}_i} \left\| \boldsymbol{D}^{J-1}\boldsymbol{\beta}_i - \boldsymbol{s}_i \right\|_2 \quad s.t. \quad \left\| \boldsymbol{\beta}_i \right\|_0 < T_0, \tag{9}$$

where $\boldsymbol{D}^{J-1}$ is the dictionary matrix obtained from the previous J–1-th iteration. Then the coefficient matrix at this iteration can be expressed as $\boldsymbol{\beta}^J = [\boldsymbol{\beta}^J{}_1, \boldsymbol{\beta}^J{}_2, ..., \boldsymbol{\beta}^J{}_N]$.

4. *Dictionary update stage*: Update the k-th basis $\boldsymbol{d}^J{}_k$, $k = 1, 2, ..., K$ in $\boldsymbol{D}^J$ by:
   - 4.1 Define the group of indices pointing to examples $\boldsymbol{S}$ that use the k-th dictionary column $\boldsymbol{d}^{J-1}{}_k$ as $\boldsymbol{\tau}_k = \{i \mid 1 \le i \le N, \beta^J{}_{T,k}(i) \ne 0\}$, where $\beta^J{}_{T,k}(i)$ is the i-th item of $\boldsymbol{\beta}^J{}_{T,k}$, and $\boldsymbol{\beta}^J{}_{T,k}$ denotes the k-th row of $\boldsymbol{\beta}^J$ but not the k-th column $\boldsymbol{\beta}^J{}_k$.
   - 4.2 Compute the representation error $\boldsymbol{E}^{J-1}{}_k$ when $\boldsymbol{d}^{J-1}{}_k$ is removed:

$$\boldsymbol{E}_k^{J-1} = \boldsymbol{S} - \sum_{\substack{j=1 \\ j \ne k}}^{K} \boldsymbol{d}_j^{J-1} \boldsymbol{\beta}_{T,j}^J. \tag{10}$$

   Then the overall representation error that should be minimized can be expressed as:

$$\left\| \boldsymbol{S} - \boldsymbol{D}^{J-1}\boldsymbol{\beta}^J \right\|_2 = \left\| \boldsymbol{S} - \sum_{i=1}^{K} \boldsymbol{d}_i^{J-1} \boldsymbol{\beta}_{T,i}^J \right\|_2 = \left\| \boldsymbol{E}_k^{J-1} - \boldsymbol{d}_k^{J-1} \boldsymbol{\beta}_{T,k}^J \right\|_2. \tag{11}$$

   - 4.3 Define a matrix $\boldsymbol{\Omega}_k \in \Re^{N \times N_k}$ with ones on $(\tau_k(i), i)$-th entries and zeroes elsewhere, where N_k is the size of $\boldsymbol{\tau}_k$ and $\tau_k(i)$ is the i-th item of $\boldsymbol{\tau}_k$. Then construct $\boldsymbol{E}^{J-1}{}_{R,k}$ by restricting $\boldsymbol{E}^{J-1}{}_k$ with $\boldsymbol{\Omega}_k$, i.e., only choosing the columns of $\boldsymbol{E}^{J-1}{}_k$ that correspond to $\boldsymbol{\tau}_k$ or the examples that use the column $\boldsymbol{d}^{J-1}{}_k$,

$$\boldsymbol{E}_{R,k}{}^{J-1} = \boldsymbol{E}_k{}^{J-1} \boldsymbol{\Omega}_k. \tag{12}$$

With this notification, the minimization of (11) is equal to minimize:

$$\left\| \boldsymbol{E}_k^{J-1}\boldsymbol{\Omega}_k - \boldsymbol{d}_k^{J-1}\boldsymbol{\beta}_{T,k}^{J}\boldsymbol{\Omega}_k \right\|_2 = \left\| \boldsymbol{E}_{R,k}^{J-1} - \boldsymbol{d}_k^{J-1}\boldsymbol{\beta}_{R,k}^{J} \right\|_2 . \quad (13)$$

4.4 Apply Singular Value Decomposition (SVD) on $\boldsymbol{E}^{J-1}{}_{R,k}$ as $\boldsymbol{E}^{J-1}{}_{R,k} = \boldsymbol{U}\Delta\boldsymbol{V}^T$. Since SVD could find the closest rank-1 matrix that approximates $\boldsymbol{E}^{J-1}{}_{R,k}$, we can update the k-th dictionary column $\boldsymbol{d}^{J}{}_{k}$ as the first column of $\boldsymbol{U}$. Then the error defined in (13) is effectively minimized. Note that in order to guarantee the sparse property of $\boldsymbol{\beta}^{J}{}_{T,k}$, SVD is applied on $\boldsymbol{E}^{J-1}{}_{R,k}$ but not on $\boldsymbol{E}^{J-1}{}_{k}$.

5. Set $J = J + 1$.

III. Robust Dictionary Learning

As we have discussed before, for the conventional VP method using general-purpose basis, the only required information for spatial variation modeling is the measurement data at some sampling locations on the wafer that needs testing. However, for a specific manufacturing process, a number of historical wafers are often available in practice. If we could take full advantage of the particular information embedded in these historical measurement data, it is very likely to improve the accuracy and/or reduce the cost of spatial variation modeling further.

Suppose that we already have L historical wafers (also called as training wafers) and the maximum number of sampling locations within each wafer is M_{max}. Note that for a given manufacturing process, the coordinates of the m-th sampling location on each wafer are often the same, where $m = 1, 2, ..., M_{max}$. Then similar as VP, the measurement performance data on each sampling location (x_m, y_m) of the l-th wafer can be expressed as $B_{l,m} = g_l(x_m, y_m)$, where $l = 1, 2, ..., L$. Let $\boldsymbol{B}_l = [B_{l,1}, B_{l,2}, ..., B_{l,Mmax}]^T$ denote the performance of l-th wafer. Then if the measurement data on all the wafers $\boldsymbol{B}_G = [\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_L]$ can be obtained rationally, we can directly apply the conventional dictionary learning algorithms (e.g., K-SVD) here to train the specific dictionary for the given manufacturing process. However, the silicon measurement data $\boldsymbol{B}_G$ are always not ideal.

First, probe misalignment could cause missing data in $\boldsymbol{B}_G$. Therefore, some elements of $\boldsymbol{B}_G$ would be unavailable. These elements, also called as missing data, are denoted as *NaN* at the rest of this paper. The positions and numbers of the missing data could vary from wafer to wafer. Obviously, it does not make sense to consider the modeling error on these missing data during spatial variation modeling. So we define a weight matrix $\boldsymbol{W}_l = diag\{u_{l,1}, u_{l,2}, ..., u_{l,Mmax}\}$, where $u_{l,i} = 1$ if $B_{l,i} \neq NaN$ and $u_{l,i} = 0$ if $B_{l,i} = NaN$. Then, the particular dictionary for the given manufacturing process that we aim to characterize, i.e., $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_K] \in \Re^{M_{max} \times K}$, can be obtained by solving,

$$\boldsymbol{W}_l(\boldsymbol{A}\boldsymbol{\eta}_l) = \boldsymbol{W}_l\boldsymbol{B}_l , \quad (14)$$

where $l = 1, 2, ..., L$, and K is the number of column basis in $\boldsymbol{A}$. For different wafer, the dictionary matrix $\boldsymbol{A}$ is the same, but the coefficient $\boldsymbol{\eta}_l$ and weight $\boldsymbol{W}_l$ are different.

Second, manufacturing defect can cause a number of measurements to be greatly deviated from the normal values, i.e., result in outliers. So directly applying the l_2-norm minimization of the modeling error as in (8) to solve (14) could easily cause misleading results. This is because the presence of the outliers would violate the assumption of Gaussian distribution for the modeling error [17]. In addition, the number of available sampling locations M_{max} might be much smaller than the number of applied basis functions K, and (14) would be underdetermined [7]. Considering the existence of outliers and the sparsity property of spatial variation, we can reformulate solving (14) as a robust regression problem:

$$\begin{aligned} \min_{\boldsymbol{\eta}_l, \boldsymbol{A}} \quad & \sum_l \left\| \boldsymbol{W}_l(\boldsymbol{A}\boldsymbol{\eta}_l - \boldsymbol{B}_l) \right\|_1 \\ s.t. \quad & \left\| \boldsymbol{\eta}_l \right\|_0 < T_0 \end{aligned} . \quad (15)$$

where $\|\boldsymbol{\eta}_l\|_0$ means the number of non-zeros in the vector $\boldsymbol{\eta}_l$, and T_0 represents its sparse level. Instead of using the common l_2-norm minimization as in (8), we apply l_1-norm of the modeling error in the cost function of (15), which could make the estimations less sensitive to the outliers. Furthermore, the employment of weight matrix $\boldsymbol{W}_l$ guarantees that the errors caused by the missing data are totally disregarded during the optimization.

In order to obtain the best dictionary $\boldsymbol{A}$ by solving (15), we propose a Robust Dictionary Learning method. This method takes use of the similar process in the conventional K-SVD algorithm, i.e., do the iterations between *sparse coding stage* and *dictionary update stage* with an initialized dictionary until convergence.

At the *sparse coding stage* of the J-th iteration, similar as K-SVD algorithm, we try to update the modeling coefficients $\boldsymbol{\eta}^{J}{}_l$, $l = 1, 2, ..., L$, for each wafer by solving the optimization problem,

$$\begin{aligned} \min_{\boldsymbol{\eta}_l} \quad & \left\| \boldsymbol{W}_l\left(\boldsymbol{A}^{J-1}\boldsymbol{\eta}_l - \boldsymbol{B}_l\right) \right\|_1 \\ s.t. \quad & \left\| \boldsymbol{\eta}_l \right\|_0 < T_0 \end{aligned} . \quad (16)$$

where $\boldsymbol{A}^{J-1}$ is the dictionary calculated at the previous $J-1$-th iteration. (16) is a sparse regression problem that could be directly solved by using any sparse regression algorithm with the l_1-norm error metric. In our implementation, orthogonal matching pursuit [16] is adopted because of its high efficiency. Further details about sparse regression methods can be founded in [14]-[15].

Then, at the *dictionary update stage*, with the given coefficient $\boldsymbol{\eta}^{J} = [\boldsymbol{\eta}^{J}{}_1, \boldsymbol{\eta}^{J}{}_2, ..., \boldsymbol{\eta}^{J}{}_L]$, we can update each column $\boldsymbol{a}^{J}{}_k$ in the dictionary $\boldsymbol{A}^{J}$ in turn by minimizing the modeling error over all the training wafers defined as,

$$\begin{aligned} \sum_l \left\| \boldsymbol{W}_l\left(\boldsymbol{B}_l - \boldsymbol{A}^{J-1}\boldsymbol{\eta}_l^{J}\right) \right\|_1 &= \sum_l \left\| \boldsymbol{W}_l\left(\boldsymbol{B}_l - \sum_{j=1}^{K} \boldsymbol{a}_j^{J-1}\eta_{l,j}^{J}\right) \right\|_1 \\ &= \sum_l \left\| \boldsymbol{W}_l\left(\boldsymbol{B}_l - \sum_{\substack{j=1 \\ j\neq k}}^{K} \boldsymbol{a}_j^{J-1}\eta_{l,j}^{J} - \boldsymbol{a}_k^{J-1}\eta_{l,k}^{J}\right) \right\|_1 \\ &= \sum_l \left\| \boldsymbol{W}_l\left(\boldsymbol{e}_{l,k}^{J-1} - \boldsymbol{a}_k^{J-1}\eta_{l,k}^{J}\right) \right\|_1 \end{aligned} , \quad (17)$$

where

$$\boldsymbol{e}_{l,k}^{J-1} = \boldsymbol{B}_l - \sum_{j=1, j\neq k}^{K} \boldsymbol{a}_j^{J-1}\eta_{l,j}^{J} . \quad (18)$$

Note that when optimizing the k-th basis $\boldsymbol{a}^{J}{}_k$, all the other basis are regarded as fixed vectors obtained from $J-1$-th iteration.

Furthermore, in order to maintain the sparse structure of the spatial variation, we would like to disregard the error $\boldsymbol{e}^{J-1}{}_{l,k}$ corresponding to the zero coefficient $\eta^{J}_{l,k}$, i.e., without using the k-th basis. Then the minimization of (18) is equivalent to minimizing,

$$\sum_{l=1}^{L}\tau_{l,k}\left\|\boldsymbol{W}_l\left(\boldsymbol{e}^{J-1}_{l,k}-\boldsymbol{a}^{J-1}_{k}\eta^{J}_{l,k}\right)\right\|_1=\sum_{l=1}^{L}\sum_{i=1}^{M_{max}}\tau_{l,k}u_{l,i}\left|e^{J-1}_{l,k,i}-a^{J-1}_{k,i}\eta^{J}_{l,k}\right|, \quad (19)$$

where $\tau_{l,k}$ is determined by the coefficient $\boldsymbol{\eta}^{J}$, i.e., $\tau_{l,k} = 1$ if $\eta^{J}{}_{l,k} \neq 0$, and $\tau_{l,k} = 0$ if $\eta^{J}{}_{l,k} = 0$. $e^{J-1}{}_{l,k,i}$ is the i-th item of $\boldsymbol{e}^{J-1}{}_{l,k}$, and $a^{J-1}{}_{k,i}$ is the i-th item of $\boldsymbol{a}^{J-1}{}_{k}$. Note that in order to avoid the misleading effects caused by missing data, only the available measurement data with $u_{l,i} \neq 0$, i.e., $e^{J}{}_{l,k,i} \neq NaN$, are taken into account. In addition, compared with (11) and (13), l_1-norm minimization of the modeling error is also used here with the consideration of outliers. Therefore, it is infeasible to directly adopt SVD as in Step 4.4 of Algorithm 1 to solve the optimization problem (19).

In order to figure out the optimized k-th basis that lead to the minimization of (19), we introduce a coordinate descent optimization method [18]. Coordinate descent optimization is based on the idea that the minimization of a multivariable function can be achieved by solving univariate optimization problems iteratively. Namely, we could minimize (19) by optimizing $\boldsymbol{a}^{J-1}{}_{k}$ and its corresponding coefficients $\boldsymbol{\eta}^{J}{}_{T,k} = [\eta^{J}{}_{1,k}, \eta^{J}{}_{2,k}, ..., \eta^{J}{}_{L,k}]$ (i.e., the k-th row of $\boldsymbol{\eta}^{J}$) iteratively as shown in Algorithm 2, where $\tau_{l,k}$, $u_{l,i}$ and $e^{J-1}{}_{l,k,i}$ are regarded to be fixed, and $\hat{\boldsymbol{a}} = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_{Mmax}]^T$ denotes $\boldsymbol{a}^{J-1}{}_{k}$ and $\hat{\boldsymbol{\eta}} = [\hat{\eta}_1, \hat{\eta}_2, ..., \hat{\eta}_L]$ denotes $\boldsymbol{\eta}^{J}{}_{T,k}$ for simplicity.

Algorithm 2: Coordinate Descent Optimization Algorithm

1. Initialize basis $\hat{\boldsymbol{a}}^0$. Set the iteration counter $t = 1$;
2. Repeat until the basis $\hat{\boldsymbol{a}}$ converges:
3. Update $\hat{\boldsymbol{\eta}}^t$ by solving the optimization problem,

$$\min_{\hat{\boldsymbol{\eta}}} \sum_{l=1}^{L}\sum_{i=1}^{M_{max}}\tau_{l,k}u_{l,i}\left|e^{J-1}_{l,k,i}-\hat{a}^{t-1}_{i}\hat{\eta}_l\right|; \quad (20)$$

   where $\hat{a}^{t-1}{}_{i}$ is the i-th item of $\hat{\boldsymbol{a}}^{t-1}$ and $\hat{\eta}_l$ is the l-th item of $\hat{\boldsymbol{\eta}}$.
4. Update $\hat{\boldsymbol{a}}^t$ by solving the optimization problem:

$$\min_{\hat{a}} \sum_{l=1}^{L}\sum_{i=1}^{M_{max}}\tau_{l,k}u_{l,i}\left|e^{J-1}_{l,k,i}-\hat{a}_i\hat{\eta}^{t}_{l}\right|; \quad (21)$$

5. Normalize $\hat{\boldsymbol{a}}^t$:

$$\hat{\boldsymbol{a}}^t=\hat{\boldsymbol{a}}^t\Big/\left(\left(\hat{\boldsymbol{a}}^t\right)^T\hat{\boldsymbol{a}}^t\right); \quad (22)$$

6. Set $t = t + 1$;

In addition, the optimization problems (20) and (21) can be formally converted to the equivalent linear programming problems due to the Karush-Kuhn-Tucker condition in the optimization theory [19]. For instance, by introducing a set of slack variables $\{\theta_{1,1}, \theta_{1,2}, ..., \theta_{L,M_{max}}\}$, (20) can be reformulated as,

$$\begin{aligned}\min\ & \sum_{i=1}^{M_{max}}\sum_{l=1}^{L}\theta_{i,l}\\ s.t.\ & \tau_{l,k}u_{l,i}\left|e^{J-1}_{l,k,i}-\hat{a}^{t-1}_{i}\hat{\eta}_l\right|\le\theta_{i,l}\\ & \theta_{i,l}\ge 0\\ & \left(i=1,2,...,M_{max},\ \text{and}\ l=1,2,...,L\right)\end{aligned} \quad . \quad (23)$$

Since both the cost function and the constraints of (23) are linear, it could be efficiently solved (e.g., by using interior point method [19]). Note that in (23), the value of the slack $\theta_{i,l}$ corresponding to $\tau_{l,k}u_{l,i} = 0$ should be zero.

Once each basis of the dictionary $\boldsymbol{A}^J$ is updated by using Algorithm 2, the Robust Dictionary Learning method would turn to the *sparse coding stage* of the next J+1-th iteration unless $\boldsymbol{A}$ is convergent.

IV. Implementation Details

A. Initialization

In order to make the proposed Robust Dictionary Learning method practically efficient, some implementation details should be considered carefully, especially the initialization of dictionary $\boldsymbol{A}$ before the iterations between *sparse coding stage* and *dictionary update stage*. Theoretically, each column of the finally learned dictionary $\boldsymbol{A}$ corresponds to one pattern of the spatial variation within the training wafers. Therefore, in order to cover as many different patterns as possible and improve the convergence rate, clustering algorithms can be utilize for the dictionary initialization. For instance, by applying the k-means clustering algorithm [20] given as Algorithm 3, where $\boldsymbol{v}_k \in \Re^{Mmax}$, $k = 1, 2, ..., K$, denote the measurement data centers of the k-th clusters, we could divide all the historical wafers into K clusters. It is rational to believe that the wafers included in the same cluster share some similar patterns. Then we select one wafer from each cluster randomly. The dictionary $\boldsymbol{A}$ can be initialized as the measurement data of these selected wafers. So the number of column basis in $\boldsymbol{A}$ and the number of divided clusters should be the same.

Algorithm 3: k-means Clustering Algorithm

1. Initialize $\{\boldsymbol{v}^0{}_1, \boldsymbol{v}^0{}_2, ..., \boldsymbol{v}^0{}_K\}$ as the measurement data of K wafers randomly selected from $\boldsymbol{B}_G$, and set the iteration counter t=1;
2. Repeat until the cluster centers $\boldsymbol{v}_k$ (k=1, 2, ..., K) converges:
3. Calculate the Euclidean distance between the measurement data of each wafer and all the cluster centers, i.e., $\|\boldsymbol{B}_l-\boldsymbol{v}^{t-1}{}_k\|_2$, where $l = 1, 2, ..., L$;
4. Assign each wafer to the cluster with the minimum distance;
5. Update the cluster center $\boldsymbol{v}^t{}_k$ as,

$$\boldsymbol{v}^t_k=\frac{1}{c_k}\sum_{i=1}^{c_k}\boldsymbol{B}_i, \quad (24)$$

   where C_k is the number of wafers included in the k-th cluster;
6. Set $t = t + 1$;

However, since there are some *NaN* items in $\boldsymbol{B}_G$ due to the existence of missing data, Algorithm 3 cannot be applied directly. Fortunately, since systematic shift dominates the wafer-to-wafer variation, the measurement data of all the wafers at this particular location, i.e., $g_l(x_m, y_m)$, $l = 1, 2, ..., L$, can be assumed to follow a Gaussian distribution [5]. So if in the w-th wafer, the measurement data at the location (x_m, y_m) is unavailable, i.e., $g_w(x_m, y_m)$ is *NaN*, we could estimate it as the

median value of $\{g_l(x_m, y_m) \mid g_l(x_m, y_m) \neq NaN, l = 1, 2, ..., L\}$. Here we choose median instead of mean, because mean value may be highly biased by the outliers [17].

In addition, for the initialization of $\hat{\boldsymbol{a}}^0$ in Algorithm 2, we can use SVD algorithm. First, let $\boldsymbol{E}^{J-1}_{R,k} = \{\boldsymbol{e}^{J-1}_{l,k} \mid \tau_{l,k} \neq 0, l = 1, 2, \ldots, L\}$ denote the errors corresponding to all the non-zero coefficients $\eta_{l,k}$. The *NaN* item in $\boldsymbol{E}^{J-1}_{R,k}$ can be filled with their column-mean value. Second, we apply SVD on the filled $\boldsymbol{E}^{J-1}_{R,k} = \boldsymbol{U}\Delta\boldsymbol{V}^T$. Then $\hat{\boldsymbol{a}}^0$ can be initialized as the first column of $\boldsymbol{U}$. Since SVD could figure out the most important basis of $\boldsymbol{E}^{J-1}_{R,k}$, the convergence rate could be rationally improved with this initialization technique.

B. Algorithm Flow

Algorithm 4 summarizes the major steps of the proposed Robust Dictionary Learning method for spatial variation modeling. Applying Algorithm 4, with the measurement data obtain from some training wafers for a particular manufacturing process, the specific dictionary can be learned by iterating between the *sparse coding stage* and *dictionary update stage*. Robust l_1-norm, rather than outlier-sensitive l_2-norm, is applied to estimate the modeling error. A coordinate descent optimization method is also introduced to deal with the missing data [18]-[19].

Once the best dictionary is learned, the spatial variation model for a new wafer that needs characterization (also called as testing wafer) could be set up easily. First, we should collect the measurement data at some sampling locations $\{(x_m, y_m); m = 1, 2, ..., M\}$ on the testing wafer. These sampling locations always constitute a small subset of the sampling locations set on the training wafers, i.e., $M << M_{max}$. Second, any sparse regression method can be applied to calculate the coefficients for the testing wafer with the learned dictionary. Note that the training wafers and the testing wafers should be fabricated from the same manufacturing process.

Algorithm 4: Robust Dictionary Learning method

1. Given a series of measurement data $\boldsymbol{B}_l$, $l = 1, 2, ..., L$ from L training wafers;
2. Formulate the weight matrix $\boldsymbol{W}_l$ for each wafer;
3. Initialize dictionary $\boldsymbol{A}^0$ by using Algorithm 3, and set the counter $J = 1$;
4. Repeat until convergence:
5. *Sparse coding stage*: Adopt any sparse regression algorithm to compute the modeling coefficients $\boldsymbol{\eta}^J_l$ for the l-th wafer with the fixed dictionary $\boldsymbol{A}^{J-1}$ by solving (16);
6. *Dictionary update stage*: Optimize the k-th basis $\boldsymbol{a}^J_k$, $k = 1, 2, ..., K$, in $\boldsymbol{A}^J$ with the fixed coefficient $\boldsymbol{\eta}^J$ by applying Algorithm 2;
7. Set $J=J+1$;

V. Numerical Examples

In this section, we validate the proposed algorithm by constructing the spatial variation models of power and frequency for a set of ring oscillators (ROs). The measurement data are collected from more than 400 wafers fabricated with the same advanced technology. On each wafer, there are 112 ROs located at different positions, i.e., $M_{max} = 112$. 300 wafers (i.e., $L = 300$) are selected randomly for dictionary learning, and another non-overlapped 100 wafers are used for testing. All the numerical experiments are performed on a computer with 3.2GHz CPU and 8GB memory.

A. Power Measurement

We validate the proposed method on the power measurement data first. Fig. 1 shows three patterns of the learned dictionary, i.e., three columns of $\boldsymbol{A}$, obtained by using the proposed method. They are represented as functions of locations (x, y).

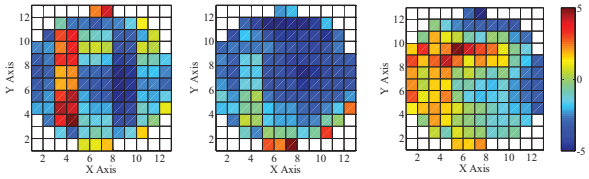

Fig. 1. Three selected basis functions from the trained dictionary are shown for the power data.

With the learned dictionary, we apply l_1-norm regularization method for the spatial variation modeling of testing wafers. For comparison, DCT basis [7] adopted by the conventional VP method are also used here to set up the model with l_1-norm error metric. In order to evaluate the modeling accuracy, we define the average modeling error as:

$$E_{AVG} = \sum_x \sum_y \left| g(x,y) - \hat{g}(x,y) \right| \Big/ \sum_x \sum_y \left| g(x,y) \right|, \qquad (25)$$

where $g(x, y)$ and $\hat{g}(x, y)$ denote the measured value and the estimated value of the power at location (x, y) on one testing wafer respectively.

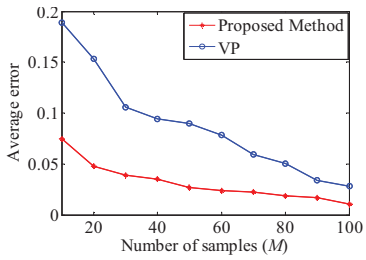

Fig. 2. The average modeling error E_{AVG} is calculated for different approaches with 100 testing wafers.

For the modeling with a given dictionary, the testing time required to generate the measurement data dominates the modeling cost of spatial variations. So the number of sampling points on one testing wafer (i.e., M) is used as a metric in this section for the comparison of modeling cost. Fig. 2 shows the mean value of the prediction error for power as a function of M with 100 testing wafers. It is clear that the proposed method achieve much better recovery accuracy than VP. For instance, when the number of sampling points is 50, the mean value of the average modeling error is 0.025 by applying the proposed method, while for VP, it is 0.090. Namely, compared with VP, the proposed method could achieve up to 70% error reduction with the similar modeling cost. This significant error reduction exactly benefits from the specific learned dictionary that carries the unique information of the particular manufacturing process.

Fig. 3 shows the modeling results of the power on one testing wafer, where (a) is the measurement data, (b) and (c)

are predicted with 40 sampling points by applying the proposed method and VP respectively. The average modeling error of the proposed method is 0.034, and the average error of VP is 0.098.

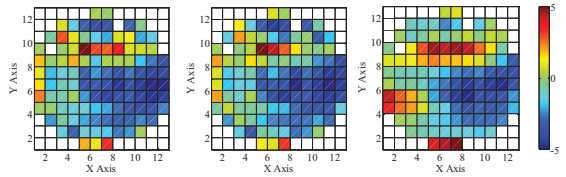

Fig. 3. The spatial variation pattern of power is plotted for a testing wafer: (a) the measurement value, (b) the spatial pattern predicted by the proposed method with 40 sampling points, and (c) the spatial pattern predicted by VP with 40 sampling points.

B. Frequency Measurement

Fig. 4 shows the modeling error of frequency as a function of the sampling points number (i.e., M) by applying different algorithms. It could also be found that the proposed method achieves better efficacy than VP. Additionally, Fig. 5 shows the modeling results of the frequency on one testing wafer. In this case, only 20 sampling points are utilized for prediction. The average modeling error of the proposed method is 0.008, and the average error of VP is 0.021.

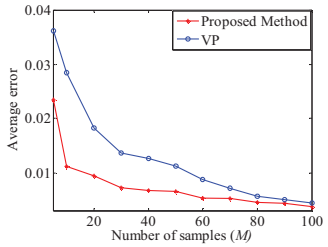

Fig. 4. The average modeling error E_{AVG} is calculated for different approaches with 100 testing wafers.

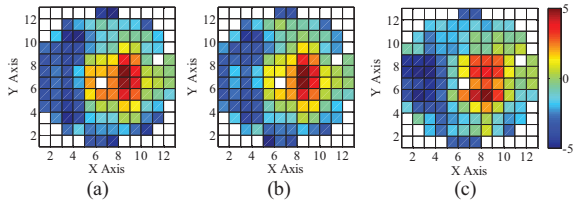

Fig. 5. The spatial variation pattern of frequency is plotted for a testing wafer: (a) the measurement value, (b) the spatial pattern predicted by the proposed method with 20 sampling points, and (c) the spatial pattern predicted by VP with 20 sampling points.

VI. Conclusion

In this paper, we propose a novel spatial variation modeling method via Robust Dictionary Learning. Applying the proposed method, a specific dictionary can be learned based on the historical wafers to capture the ad-hoc patterns of a particular manufacturing process. In addition, robust l_1-norm is applied for the minimization of modeling error in order to reduce the misleading effects of outliers. Then an efficient coordinate descent method is introduced to learn the dictionary with the consideration of missing data. Numerical results demonstrate that compared with VP, the proposed method could achieve up to 70% accuracy improvement with the similar low measurement cost.

Acknowledgement

This research is supported partly by National Natural Science Foundation of China (NSFC) research project 61376041, 61125401, 61376040, 91330201, 61574046, and 61574044, partly by the National Basic Research Program of China under the grant 2011CB309701, partly by the Recruitment Program of Global Experts (the Thousand Talents Plan), partly by National Science Foundation (NSF) research project 1115556 and CCF–1316363, partly by the Laboratory of Mathematics for Nonlinear Science, Fudan University.

References

[1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2013. [Online]. http://www.itrs.net/Links/2013ITRS/Home2013.htm

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V.De, "Parameter variations and impact on circuits and microarchitecture," *DAC*, pp. 338-342, Jun. 2003.

[3] X. Li, R. Rutenbar and R. Blanton, "Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," *ICCAD*, pp. 433–440, 2009.

[4] W. Zhang, X. Li and R. Rutenbar, "Bayesian virtual probe: minimizing variation characterization cost for nanoscale IC technologies via Bayesian inference," *DAC*, pp. 262-267, 2010.

[5] W. Zhang, X. Li, E. Acar, F. Liu and R. Rutenbar, "Multi-wafer virtual probe: minimum-cost variation characterization by exploring wafer-to-wafer correlation," *ICCAD*, pp. 47-54, 2010.

[6] H. Chang, K. Cheng, W. Zhang, X. Li and K. Butler, "Test cost reduction through performance prediction using virtual probe," *ITC*, 2011.

[7] W. Zhang, X. Li, T, Liu, E. Acar, R. R. Rutenbar and R. D. Blanton, "Virtual probe: a statistical framework for low-cost silicon characterization of nanoscale integrated circuits," *IEEE TCAD*, vol. 30, no. 12, pp. 1814-1827, 2011.

[8] W. Zhang, X. Li, S. Saxena, A. Strojwas and R. Rutenbar, "Automatic clustering of wafer spatial signatures," *DAC*, 2013.

[9] W. Zhang, K. Balakrishnan, X. Li, D. Boning, S. Saxena, A. Strojwas and R. Rutenbar, "Efficient variation decomposition via robust sparse regression," *IEEE TCAD*, vol. 32, no. 7, pp. 1072-1085, Jul. 2013.

[10] H. G., X. Li, M. Correia, V. Tavares, J. Carulli and K. Butler, "A fast spatial variation modeling algorithm for efficient test cost reduction of analog/RF circuits," *DATE*, pp. 1042-1047, 2015.

[11] N. Kupp, K. Huang, J. Carulli and Y. Makris, "Spatial estimation of wafer measurement parameters using Gaussian process models," *ITC*, 2012.

[12] K. Kreutz-Delgado *et al.*, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349-396, 2003.

[13] M. Aharon, M. Elad, and A. Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE TSP*, vol. 54, no. 11, pp. 4311-4322, Sep. 2006.

[14] D. Donoho, "Compressed sensing," *IEEE TIT*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[15] E. J. Candes, "Compressive sampling," *International Congress of Mathematicians*, vol. 3, pp. 1433–1452, 2006.

[16] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE TIT*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[17] R. Maronna, R. Martin, and V. Yohai, *Robust Statistics: Theory and Methods*, John Wiley and Sons, 2006.

[18] R. Fletcher, *Practical methods of optimization*. NY: John Wiley & Sons, 2013.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

[20] C. Bishop, *Pattern Recognition and Machine Learning*. Upper Saddle River, NJ: Prentice-Hall, 2007.