Efficient Spatial Pattern Analysis for Variation Decomposition via Robust Sparse Regression

Wangyang Zhang, *Member, IEEE,* Karthik Balakrishnan, Xin Li, *Senior Member, IEEE,* Duane S. Boning, *Fellow, IEEE,* Sharad Saxena, *Senior Member, IEEE,* Andrzej Strojwas, *Fellow, IEEE,* and Rob A. Rutenbar, *Fellow, IEEE*

***Abstract*—In this paper, we propose a new technique to achieve accurate decomposition of process variation by efficiently performing spatial pattern analysis. We demonstrate that the spatially correlated systematic variation can be accurately represented by the linear combination of a small number of templates. Based on this observation, an efficient sparse regression algorithm is developed to accurately extract the most adequate templates to represent spatially correlated variation. In addition, a robust sparse regression algorithm is proposed to automatically remove measurement outliers. We further develop a fast numerical algorithm that may reduce the computational time by several orders of magnitude over the traditional direct implementation. Our experimental results based on both synthetic and silicon data demonstrate that the proposed sparse regression technique can capture spatially correlated variation patterns with high accuracy and efficiency.**

***Index Terms*—Integrated circuit, process variation, spatial variation, variation decomposition.**

I. Introduction

With the continued scaling of CMOS technology, process variation has become a critical issue for the design and manufacturing of integrated circuits [1]. Large-scale performance variability has been observed for integrated circuits fabricated at advanced technology nodes, resulting in significant parametric yield loss. For this reason, accurate process characterization and modeling is required to fully understand the sources of variation.

Variation decomposition is an important tool to achieve this goal. Different variation components indicate different physical variation sources. From a geometrical perspective, process variation can be decomposed into lot-to-lot variation, wafer-to-wafer variation, wafer-level variation, and within-die variation. Once the geometrical level of variation is known, it narrows down the underlying variation sources [2]. To further narrow down the physical sources of variation, we would like to further decompose process variation into systematic and random components. It has been demonstrated in the literature that systematic variation often presents a unique spatial pattern [2]. Namely, systematic variation is often spatially correlated. For example, it has been observed in [4] that the spatial correlation in gate length is partially caused by the systematic variation due to lithography.

A number of prior works [2]–[6] have been proposed for modeling spatially correlated variation. Some of them, such as [6], represent the spatially correlated variation as random variables, and their correlation is modeled as a function of distance. These methods do not fit the needs of our applications for variation decomposition, because they do not explicitly separate spatially correlated systematic variation from random variation. Other works [2]–[5] model the spatially correlated systematic variation by a small number of predetermined templates (e.g., linear and quadratic functions). However, the optimal templates for spatial variation modeling may vary over different processes and/or designs. If a few templates are considered, the spatially correlated systematic variation cannot be accurately captured. On the other hand, applying a large number of templates also leads to inaccurate modeling results due to overfitting [25].

Motivated by these observations, we propose a novel sparse regression technique to perform spatial pattern analysis. Our goal is to accurately model spatially correlated systematic variation and separate it from uncorrelated random variation. To apply sparse regression, only a dictionary of templates is needed, which includes all possible patterns of spatially correlated systematic variation. The optimal templates to model the spatially correlated variation of a given wafer/die will be automatically selected by the sparse regression algorithm. We have constructed a physical dictionary that contains a number of templates based on common physical variation sources. To model variation sources that are not covered by the physical dictionary, a general dictionary containing discrete cosine transform (DCT) [26] functions can be further applied to complement the physical dictionary because of the unique sparse structure of spatially correlated variation

Manuscript received August 23, 2012; revised November 8, 2012 and January 9, 2013; accepted January 22, 2013. Date of current version June 14, 2013. This work was supported in part by the C2S2 Focus Center, the Interconnect Focus Center, and the National Science Foundation under Grant CCF-0915912. This paper was recommended by Associate Editor S. Vrudhula.

W. Zhang was with Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Cadence Design Systems, Inc., Pittsburgh, PA 15238 USA (e-mail: zwy677@gmail.com).

K. Balakrishnan was with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: kbalakr@us.ibm.com).

X. Li and A. Strojwas are with Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: xinli@ece.cmu.edu; ajs@ece.cmu.edu).

D. Boning is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: boning@mtl.mit.edu).

S. Saxena is with PDF Solutions, Inc., Richardson, TX 75082 USA (e-mail: sharad.saxena@pdf.com).

R. A. Rutenbar is with the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: rutenbar@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2013.2245942

in the frequency domain [7]–[9]. After the optimal templates are selected, they are provided to a linear mixed model [24] to perform variation decomposition using standard variance components techniques.

Another important contribution of this paper is to develop a robust solver to accurately select the templates and remove measurement outliers. Silicon measurement data are usually error prone. A substantial error can be introduced to variation decomposition if outliers are not appropriately considered. To address this issue, we borrow the simultaneous orthogonal matching pursuit (S-OMP) method from the statistics community [10] and further make it insensitive to outliers based on the concept of robust regression [27]. Several new numerical algorithms are further developed to substantially reduce the computational time of the proposed solver for large-scale wafer/chip-level data analysis.

The proposed variation decomposition technique has been validated by several synthetic data sets and silicon measurement data from advanced CMOS processes. As will be demonstrated by the experimental results in Section VI, the proposed variation decomposition technique accurately models the spatially correlated systematic variation in the presence of outliers. In addition, our improved algorithm implementation achieves about 200× speedup over the traditional S-OMP implementation for a large-scale problem.

The remainder of this paper is organized as follows. In Section II, we set up the mathematical formulation for the variation decomposition problem. Next, we introduce the dictionaries of templates in Section III and describe the robust S-OMP algorithm in Section IV. We develop several numerical techniques to implement the robust S-OMP algorithm in Section V. The efficacy of the proposed method is demonstrated by the examples in Section VI. Finally, we conclude in Section VII.

II. Variation Decomposition

To decompose process variation from a geometrical perspective, the overall variation can be first represented by the following nested model [3]:

$$b_{lkji} = \tau_l + \theta_{k(l)} + \gamma_{j(kl)} + \varepsilon_{i(jkl)} \tag{1}$$

where b_{lkji} indicates the overall variation, τ_l is the lth lot variation, $\theta_{k(l)}$ is the kth wafer variation within the lth lot, $\gamma_{j(kl)}$ is the jth die variation within the kth die and lth wafer, and finally $\varepsilon_{i(jkl)}$ is the ith within-die variation within the jth die, the kth wafer, and the lth lot. For wafer-level and within-die variation, we further extract the spatially correlated variation by the linear combination of a set of predefined basis functions

$$\gamma_{j(kl)} = \sum_{m=1}^{\lambda_1} A_{wafer,m}(x_{die,j}, y_{die,j}) \cdot \alpha_m + \gamma^r_{j(kl)} \tag{2}$$

$$\varepsilon_{i(jkl)} = \sum_{m=1}^{\lambda_2} B_{die,m}(x_{site,i}, y_{site,i}) \cdot \beta_m + \varepsilon^r_{i(jkl)}. \tag{3}$$

Each basis function can be viewed as a particular template to model the spatially correlated variation. The wafer-level spatially correlated variation is represented by λ_1 basis functions $\{A_{wafer,m}(x_{die,j}, y_{die,j}); m = 1, 2, \ldots, \lambda_1\}$, where $(x_{die,j}, y_{die,j})$ is the location of the jth die on the wafer, and $\gamma^r_{j(kl)}$ represents the wafer-level random variation. Similarly, the within-die spatially correlated variation is represented by λ_2 basis functions $\{A_{die,m}(x_{site,i}, y_{site,i}); m = 1, 2, \ldots, \lambda_2\}$, where $(x_{site,i}, y_{site,i})$ is the location of the ith measurement site on the die, and $\varepsilon^r_{i(jkl)}$ represents the within-die random variation.

By combining (1)–(3), we obtain the following representation of the overall variation:

$$\begin{aligned} b_{lkji} = {} & \tau_l + \theta_{k(l)} + \textstyle\sum_{m=1}^{\lambda_1} A_m(x_{die,j}, y_{die,j}) \cdot \alpha_m + \gamma^r_{j(kl)} \\ & + \textstyle\sum_{m=1}^{\lambda_2} B_m(x_{site,i}, y_{site,i}) \cdot \beta_m + \varepsilon^r_{i(jkl)} \end{aligned} \tag{4}$$

where the overall variation is decomposed into six components: lot-to-lot variation, wafer-to-wafer variation, wafer-level spatially correlated variation, wafer-level random variation, within-die spatially correlated variation, and within-die random variation. Equation (4) is referred to as a linear mixed model [24] in statistics, which is most commonly estimated using the restricted maximum likelihood (REML) method [24], yielding the coefficients $\{\alpha_m; m = 1, 2, \ldots, \lambda_1\}$ and $\{\beta_m; m = 1, 2, \ldots, \lambda_2\}$ for wafer-level and within-die spatially correlated variation, and the variances σ^2_{lot}, σ^2_{wafer}, $\sigma^2_{die,r}$ and $\sigma^2_{site,r}$ for lot-to-lot, wafer-to-wafer, wafer-level random, and within-die random variation, respectively. In order to compare the contributions of spatially correlated variations with the random variation components, we also estimate the variance for spatially correlated wafer-level and within-die variation by the following sample variance estimation:

$$\sigma^2_{die,s} = var\left(\left\{\sum_{m=1}^{\lambda_1} A_m(x_{die,j}, y_{die,j}) \cdot \alpha_m; j = 1, 2, \ldots, N_{die}\right\}\right) \tag{5}$$

$$\sigma^2_{site,s} = var\left(\left\{\sum_{m=1}^{\lambda_2} B_m(x_{site,i}, y_{site,i}) \cdot \beta_m; i = 1, 2, \ldots, N_{site}\right\}\right) \tag{6}$$

where var(•) stands for the sample variance, N_{die} is the number of dies on the wafer, and N_{site} is the number of measurement sites in a die. The contribution of a particular component is then estimated by dividing its variance value with the sum of variance for all components. This allows the process engineers to prioritize their goals in improving yield and focus their efforts on variation components that have stronger impact on overall variability. Note that, in practice, due to the limitation of measurements, we may only be able to estimate part of these variance values. For example, in early-stage yield learning, there may be only one wafer and only a single performance value is obtained from each die. In this case, we are only able to extract the wafer-level spatially correlated and random components. The contribution of wafer-level spatially correlated variation can be calculated by $\sigma^2_{die,s}/(\sigma^2_{die,s} + \sigma^2_{die,r})$, and the contribution of wafer-level random variation can be calculated by $\sigma^2_{die,r}/(\sigma^2_{die,s} + \sigma^2_{die,r})$.

An important problem in applying the linear mixed model (4) is that the appropriate basis functions must be selected to model the spatially correlated variation. Traditionally, only a small number of simple linear or quadratic basis functions are employed [2]–[5]. These simple basis functions are only

capable of modeling a limited amount of variation sources and are not sufficient for modern manufacturing processes. For example, an important problem of modern processes is that edge dies on a wafer can have significantly lower yield than other parts of the wafer [1]. Since the outer 20 mm of a 300-mm wafer can contain up to 25% of the dies on a wafer [13], the aforementioned edge effect can lead to substantial impact to the overall yield. Therefore, nontrivial basis functions are needed to capture the systematic variation sources, such as those related to the edge effect. In Section III, we will propose our basis function dictionaries to address this issue.

III. Basis Function Dictionaries

For both wafer-level variation and within-die variation, we need to construct a dictionary of basis functions to capture the spatial patterns that can be produced by a large number of potential physical effects. In this subsection, we propose two possible dictionaries of basis functions. The first dictionary includes basis functions based on actual physical effects, and the second dictionary is constructed by the basis functions from DCT.

A. Physical Dictionary

We first introduce the physical basis function dictionary. For the sake of simplicity, we simply use x and y to designate the spatial location of a die on a wafer or a measurement site within a die. The actual physical meaning of x and y will be explained in the context.

It has been shown that a large number of wafer-level and within-die physical effects can be modeled using a quadratic model of x and y

$$f_{quad}(x, y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 y^2 + a_6 xy. \quad (7)$$

This model corresponds to six physical basis functions: $\{1,\ x,\ y,\ x^2, y^2, xy\}$. A quadratic wafer-level pattern has been observed for a large number of physical effects, such as post-exposure baking temperature related critical dimension (CD) variation [14], etch temperature related CD variation [14], and deposition rate variation of chemical vapor deposition [15]. It is shown in [4] that within-die gate CD variation can be modeled using a quadratic function, and such a pattern can be explained by the along-slit and along-scan variation of the scanner [21]. Therefore, the quadratic basis functions are included in the physical dictionary of both wafer-level and within-die variation.

For wafer-level variation associated with several process steps such as etching [13], [16] and rapid thermal annealing [17], [18], it is observed that edge dies are often substantially different from other parts of the wafer [13], in addition to the effects that can be modeled using a quadratic function. We capture the edge effect by supplementing quadratic functions with the following indicator functions:

$$f_{edge}(x, y) = \begin{cases} 1 & (x, y) \in E \\ 0 & otherwise \end{cases} \quad (8)$$

where E is a predefined subset of dies that belong to the edge region of a wafer. For example, an edge basis function can be intuitively defined according to Fig. 1(a), where a die is considered an edge die if one or more of its neighbors are not a valid die on the wafer. Since the edge effect may affect multiple layers of dies, we define the edge dies corresponding to Fig. 1(a) as the depth 1 edge of a wafer, and recursively define that a die belongs to the depth i edge of a wafer, if itself or one of its neighbors belongs to the depth $i-1$ edge of a wafer. An example of the depth 2 edge of the same wafer is shown in Fig. 1(b). Edge basis functions with different depths can be included in the physical dictionary, and the actual basis function that optimally matches a particular process can be automatically selected by the sparse regression algorithm (i.e., Algorithm 2) in Section IV.

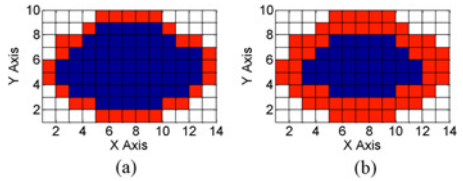


Fig. 1. (a) Depth 1 edge of a wafer and (b) depth 2 edge of a wafer, where edge dies are marked in red.

At the nwafer level, the edge effect is nonuniform; we may only observe the edge effect from a portion of the edge dies, and edge effects at different regions of a wafer can be different. In order to accurately capture the edge effect under such nonuniformity, we further partition the edge dies of a wafer into multiple regions, and construct an individual basis function for each region of the wafer. For example, two different methods to partition the depth 1 edge into four regions are shown in Fig. 2, yielding eight basis functions in total. Similar partitioning can be performed for other depths, and the resulting basis functions are all included in the physical basis function dictionary.

Other than the quadratic and edge effects, the center region of a wafer can be significantly different from other parts of the wafer. Such a center effect can occur due to several variation sources, e.g., photoresist spinning and ion implantation. We construct the following indicator functions for the center effect:

$$f_{\text{center}}(x, y) = \begin{cases} 1, & (x, y) \in C \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where C is a predefined subset of dies that belong to the center region of a wafer. Since it is difficult to uniquely define the center region in advance, we include multiple basis functions that correspond to different center definitions. For example, Fig. 3 shows four different definitions of the center region for a 13×9 wafer. If the center effect exists, the most suitable basis function will be automatically selected from these candidates by the sparse regression algorithm (i.e., Algorithm 2) in Section IV.

For within-die variation, measurements may be collected from test structures with different layouts to capture the layout-dependent variation. One possible method of modeling layout-dependent variation is again by the indicator functions

$$f_{L_i}(x, y) = \begin{cases} 1, & (x, y) \in L_i \\ 0, & \text{otherwise} \end{cases} \quad (i = 1, 2, \ldots, N) \quad (10)$$

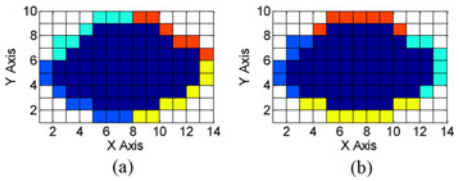


Fig. 2. (a), (b) Two different methods that partition the depth 1 edge into four regions.

where L_i is a set of measurements collected from the test structures with the layout style i. Suppose that there are N different layout styles for the within-chip test structures, N different basis functions can be used to model the systematic difference in performance caused by different layout styles.

B. DCT Dictionary

While several physical variation sources can be modeled by the previously defined physical dictionary, it is by no means complete in modeling all variation sources for today's complicated manufacturing process. More physical basis functions can be defined and included in the physical basis function dictionary, once we understand more physical variation sources. On the other hand, another dictionary of DCT basis functions borrowed from the image processing literature can be applied to complement the physical dictionary to model the variation sources that are not well understood. In what follows, we will first construct the DCT dictionary, and then explain why it can be used to decompose spatially correlated and random variation.

Let $b(x, y)$ be a 2-D function representing the spatial variation of interest, where x and y denote the coordinate of a spatial location within the 2-D plane. In practice, the spatial variation b is obtained from measurements at a finite number of spatial locations. Without loss of generality, the spatial coordinates x and y can be labeled as integer numbers: $x \in \{1, 2, ..., P\}$ and $y \in \{1, 2, ..., Q\}$, as shown in [7]–[9]. If the spatial variation b is obtained from multiple wafers and/or dies, it can be represented by a set of 2-D functions: $\{b_{(l)}(x, y); l = 1, 2, \ldots, L\}$, where L denotes the total number of wafers/dies. The DCT is a 2-D orthogonal linear transform that maps the spatial variation $\{b_{(l)}(x, y); x = 1, 2, ..., P, y = 1, 2, ..., Q\}$ to the frequency domain [26]

$$D_{(l)}(u, v) = \sum_{x=1}^{P}\sum_{y=1}^{Q} \alpha_u \cdot \beta_v \cdot b_{(l)}(x, y) \cdot \cos\frac{\pi(2x-1)\cdot(u-1)}{2\cdot P} \cdot \cos\frac{\pi(2y-1)\cdot(v-1)}{2\cdot Q} \quad (l = 1, 2, \ldots, L) \tag{11}$$

where

$$\alpha_u = \begin{cases} \sqrt{1/P} & (u = 1) \\ \sqrt{2/P} & (2 \le u \le P) \end{cases}, \beta_v = \begin{cases} \sqrt{1/Q} & (v = 1) \\ \sqrt{2/Q} & (2 \le v \le Q). \end{cases} \tag{12}$$

In (11), $\{D_{(l)}(u, v); u = 1, 2, ..., P, v = 1, 2, ..., Q\}$ represents the DCT coefficients (i.e., the frequency-domain components) of the spatial variation function $b_{(l)}(x, y)$. Equivalently, the function $\{b_{(l)}(x, y); x = 1, 2, ..., P, y = 1, 2, ..., Q\}$ can be represented as the linear combinations of $\{D_{(l)}(u, v); u = 1, 2, ..., P, v = 1, 2, ..., Q\}$ by applying inverse discrete cosine transform

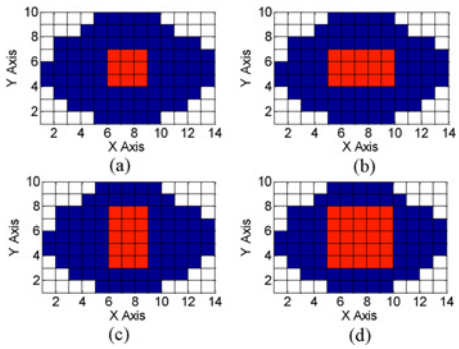


Fig. 3. (a)–(d) Four different center region definitions of the same wafer, where center dies are marked in red.

$$b_{(l)}(x, y) = \sum_{u=1}^{P}\sum_{v=1}^{Q} \alpha_u \cdot \beta_v \cdot D_{(l)}(u, v) \cdot \cos\frac{\pi(2x-1)(u-1)}{2\cdot P} \cdot \cos\frac{\pi(2y-1)(v-1)}{2\cdot Q} \quad (l = 1, 2, \ldots, L). \tag{13}$$

We construct the DCT dictionary by including the PQ basis functions in (13).

Next, we will explain why the decomposition of spatially correlated variation and random variation can be achieved by using the DCT dictionary. We first decompose $b_{(l)}(x, y)$ by the following equation:

$$b_{(l)}(x, y) = s_{(l)}(x, y) + r_{(l)}(x, y) \quad (l = 1, 2, \ldots, L) \tag{14}$$

where $s_{(l)}(x, y)$ and $r_{(l)}(x, y)$ stand for the spatially correlated variation and the uncorrelated random variation, respectively. Due to the linearity of DCT [26], the decomposition (14) can be equivalently performed in the frequency domain

$$D_{(l)}(u, v) = S_{(l)}(u, v) + R_{(l)}(u, v) \quad (l = 1, 2, \ldots, L) \tag{15}$$

where $S_{(l)}(u, v)$ and $R_{(l)}(u, v)$ denote the DCT coefficients of the spatially correlated variation $s_{(l)}(x, y)$ and the uncorrelated random variation $r_{(l)}(x, y)$ defined in (14). Once $S_{(l)}(u, v)$ and $R_{(l)}(u, v)$ are found, $s_{(l)}(x, y)$ and $r_{(l)}(x, y)$ can be determined by IDCT. As is demonstrated in [7]–[9], the DCT coefficients $S_{(l)}(u, v)$ (corresponding to spatially correlated variation) are typically sparse, i.e., many of these coefficients are close to 0. In other words, there exist a small number of (say, λ where $\lambda << PQ$) dominant DCT coefficients to satisfy

$$\sum_{(u,v)\in\Omega} S_{(l)}^2(u, v) \approx \sum_{u=1}^{P}\sum_{v=1}^{Q} S_{(l)}^2(u, v) \tag{16}$$

where Ω denotes the set of the indices of the dominant DCT coefficients for $S_{(l)}(u, v)$. Equation (16) simply implies that the total energy of all DCT coefficients $\{S_{(l)}(u, v); u = 1, 2, ..., P, v = 1, 2, ..., Q\}$ are almost equal to the energy of the dominant DCT coefficients $\{S_{(l)}(u, v); (u, v) \in \Omega\}$. On the other hand, uncorrelated random variation can be characterized as white noise [26]

and evenly distributed among all frequencies. Hence, given the set of indices Ω, the following equation holds:

$$\sum_{(u,v)\in\Omega} R_{(l)}^2(u,v) \approx \frac{\lambda}{PQ} \cdot \sum_{u=1}^{P}\sum_{v=1}^{Q} R_{(l)}^2(u,v). \tag{17}$$

Because of the inequality $\lambda << PQ$, we have $\lambda/PQ << 1$ in (17). If the value of λ is sufficiently small (i.e., the DCT coefficients of spatially correlated variation are sufficiently sparse), the left-hand side of (17) is approximately zero and the following inequality holds:

$$\sum_{(u,v)\in\Omega} R_{(l)}^2(u,v) << \sum_{(u,v)\in\Omega} S_{(l)}^2(u,v). \tag{18}$$

Based on (16) and (18), the DCT coefficients $S_{(l)}(u, v)$ (corresponding to spatially correlated variation) can be simply approximated by the dominant DCT coefficients $\{D_{(l)}(u, v); (u, v) \in \Omega\}$ with a negligible error. Applying the DCT dictionary requires knowing the set Ω of the indices of dominant DCT coefficients. This set can be identified by using the sparse regression method introduced in the next section.

It should be noted that other dictionaries that offer sparsity for spatially correlated variation, such as the wavelet basis functions [26], can be adopted as well to complement the physical dictionary. However, we find that DCT often outperforms wavelet when modeling spatial variation patterns. The fundamental reason is because wavelet basis functions are localized in the spatial domain. In practice, many physical sources of process variation will impact the whole wafer and/or die. To model these variation sources with long correlation distance, DCT can be more effective than wavelet.

There are two dictionaries proposed in this section. Since the physical dictionary is constructed from actual physical sources, it provides useful insights for process engineers. Therefore, as will be discussed in the next section, we will prioritize the physical dictionary over the DCT dictionary when selecting the basis functions. On the other hand, while the DCT basis functions do not have clear physical meaning, it is still possible to identify the variation sources by inspecting the spatial pattern represented by DCT basis functions and comparing it with those produced by various process steps/equipments. The DCT dictionary contains a large number of basis functions, which greatly increases the algorithm complexity. We will discuss this issue in detail in Section V.

IV. Robust S-OMP Algorithm

In this section, we will first formulate a sparse regression problem for basis function selection and then describe the robust S-OMP algorithm to solve the problem.

A. Basis Selection via Sparse Regression

In the previous section, we have developed two dictionaries that contain a large number of possible basis functions to model spatially correlated variation. For a particular process or design, the actual basis functions should be selected from the dictionaries to achieve accurate modeling and avoid overfitting. In this subsection, we will show that this basis selection problem can be solved by applying sparse regression.

It has been shown in (14) that for any wafer or die, the spatial variation $b_{(l)}(x, y)$ can be represented as the summation of the spatially correlated variation $s_{(l)}(x, y)$ and the random variation $r_{(l)}(x, y)$. We first model the spatially correlated variation using the basis functions from the physical dictionary. The physical dictionary is prioritized over the DCT dictionary, because its basis functions carry the physical meaning that can be further utilized to analyze the physical variation sources. To this end, we represent $s_{(l)}(x, y)$ as a linear combination of all basis functions from the physical dictionary

$$b_{(l)}(x,y) = \sum_{j=1}^{M_{phys}} \eta_{phys(l),j} \cdot A_{phys,j}(x,y) + r_{(l)}(x,y) \tag{19}$$

where the spatially correlated variation is represented by all M_{phys} basis functions $\{A_{phys,j}(x, y); j = 1, 2, \ldots, M_{phys}\}$ in the physical dictionary with the coefficients $\{\eta_{phys(l),j}; j = 1, 2, \ldots, M_{phys}\}$. Since we aim to identify the subset of basis functions that are relevant to a particular process/design, the coefficients are required to be sparse. In other words, many coefficients must be 0 in (19).

To solve the model in (19), the spatial variation $b_{(l)}(x, y)$ is measured at a finite number of spatial locations $\{b_{(l)}(x_i, y_i); i = 1, 2, \ldots, N_{(l)}\}$, and we want to estimate the sparse coefficients $\{\eta_{phys(l),j}; j = 1, 2, \ldots, M_{phys}\}$ from such measurement data. Therefore, we formulate the sparse regression problem

$$\begin{aligned} \underset{\eta_{phys_{(l)}}}{\text{minimize}} \quad & \left\| A_{phys(l)}\eta_{phys(l)} - B_{(l)} \right\|_2^2 \quad (l = 1, 2, \ldots, L) \\ \text{s.t.} \quad & \left\| \eta_{phys(l)} \right\|_0 \le \lambda_{phys} \end{aligned} \tag{20}$$

where $B_{(l)} = [b_{(l)}(x_1, y_1)\ b_{(l)}(x_2, y_2)\ \ldots b_{(l)}(x_{N_{(l)},S}, y_{N_{(l)s}})]^T$ is a vector of spatial variation measurements, $\eta_{phys(l)} = [\eta_{phys(l),1}\ \eta_{phys(l),2}\ \ldots \eta_{phys(l),M_{phys}}]^T$ is a vector of coefficients for physical basis functions, and $A_{phys(l)}$ is a matrix where $A_{phys(l),ij}$ represents the value of the jth physical basis function at the ith measurement location. The symbol $||\bullet||_2$ stands for the L_2-norm (i.e., the square root of the summation of the squares of all elements) of a vector, and $||\bullet||_0$ stands for the L_0-norm (i.e., the number of nonzeros) of a vector. The cost function indicates that we would like to fit the measurement data with least-squares error. On the other hand, the constraint controls the sparsity of $\eta_{(l)}$, which means that out of all possible M_{phys} candidates in the dictionary, there exists a small subset of λ_{phys} basis functions that are applied to model the spatially correlated variation. Therefore, the meaning of (20) is to optimally select λ_{phys} basis functions to model the spatially correlated variation. The numerical solver for the problem will be discussed in the next subsection. The optimization (20) is solved to select the basis functions for wafer-level and within-die spatially correlated variations respectively, and then the linear mixed model (4) is formulated using these selected basis functions and solved using the REML method.

As discussed in Section III-B, it may not be sufficient to model the spatially correlated variation using the physical basis functions only. Therefore, the DCT dictionary is used to

complement the physical dictionary and check if there is any significant spatial pattern that has been missed by the physical basis functions. Toward this goal, we further represent $s_{(l)}(x, y)$ as a linear combination of the selected physical basis functions and all DCT basis functions

$$b_{(l)}(x, y) = \sum_{j \in \Omega_{phys}} \eta_{phys(l),j} \cdot A_{phys,j}(x, y) + \sum_{j=1}^{M_{dct}} \eta_{dct(l),j} \cdot A_{dct,j}(x, y) + r_{(l)}(x, y) \tag{21}$$

where Ω_{phys} represents the subset of physical basis functions selected by solving (20). In addition, the spatially correlated variation is further represented by all basis functions $\{A_{dct,j}(x, y); j = 1, 2, \ldots, PQ\}$ in the DCT dictionary with coefficients $\{\eta_{dct(l),j}; j = 1, 2, \ldots, PQ\}$. Again, the DCT coefficients are required to be sparse, leading to the following sparse regression problem:

$$\begin{aligned} \underset{\eta_{phys_{(l),\Omega_{phys}}}, \eta_{dct(l)}}{\text{minimize}} \quad & \left\| A_{phys(l),\Omega_{phys}} \eta_{phys(l),\Omega_{phys}} + A_{dct(l)} \eta_{dct(l)} - B_{(l)} \right\|_2^2 \\ \text{s.t.} \quad & \left\| \eta_{dct(l)} \right\|_0 \leq \lambda_{dct} \quad (l = 1, 2, \ldots, L) \end{aligned} \tag{22}$$

where $A_{phys(l),\Omega_{phys}}$ is a submatrix of $A_{phys(l)}$ containing the columns that belong to Ω_{phys}, $\eta_{phys(l)}$, Ω_{phys} is a vector of the corresponding coefficients for physical basis functions, $\eta_{dct(l)} = [\eta_{dct(l),1} \eta_{dct(l),2} \ldots \eta_{dct(l),PQ}]^T$ is a vector of coefficients for all DCT basis functions, and $A_{phys(l)}$ is a matrix where $A_{phys(l),ij}$ represents the value of the jth DCT basis function at the ith measurement location. The optimization (22) optimally selects λ_{dct} DCT basis functions to model the spatially correlated variation. The linear mixed model (4) is then formulated using the selected basis functions and solved using the REML method.

B. Simultaneous Orthogonal Matching Pursuit

In the previous subsection, we have formulated two sparse regression problems (20) and (22) to select the basis functions to represent the spatially correlated variation. Both problems follow the general mathematical formulation:

$$\begin{aligned} \underset{\eta_{(l)}}{\text{minimize}} \quad & \left\| A_{(l)} \eta_{(l)} - B_{(l)} \right\|_2^2 \\ \text{s.t.} \quad & nnz\left(\left\{ \eta_{(l),j} | j \notin \Omega_0 \right\} \right) \leq \lambda \end{aligned} \quad (l = 1, 2, \ldots, L) \tag{23}$$

where Ω_0 represents a set of basis functions that are preselected, and nnz$(\cdot)$ stands for the number of nonzeros within a set. Equation (20) is a special case of (23) where $A_{(l)} = A_{phys(l)}$, $\eta_{(l)} = \eta_{phys(l)}$, $\lambda = \lambda_{phys}$ and $\Omega_0 = \emptyset$. Equation (22) is a special case of (23) where

$$A_{(l)} = \begin{bmatrix} A_{phys(l),\Omega_{phys}} & A_{dct(l)} \end{bmatrix} \tag{24}$$

$$\eta_{(l)} = \begin{bmatrix} \eta_{phys(l),\Omega_{phys}} \\ \eta_{dct(l)} \end{bmatrix} \tag{25}$$

$\lambda = \lambda_{dct}$ and $\Omega_0 = \{1, 2, \ldots, \lambda_{phys}\}$. In general, solving the optimization (23) is not trivial, since the problem is NP-hard. Several efficient solvers for (23) have been discussed in the statistics literature, such as group lasso (GL) [11] and S-OMP [10]. We select S-OMP as the numerical solver for the sparse regression problem in this paper. S-OMP is a greedy algorithm to approximate the solution of (23) when $\Omega_0 = \emptyset$. An important reason for choosing S-OMP is its simplicity, which allows us to easily adapt the algorithm for several practical needs, such as predetermined basis functions, outlier detection and fast computation with the DCT dictionary. In what follows, we will first briefly review the major steps of the S-OMP algorithm. Then, we will further extend the S-OMP algorithm to solve (23) when Ω_0 is nonempty.

The key idea of S-OMP is to iteratively use the inner product to identify a small number of important basis functions. To this end, we rewrite the matrix $A_{(l)}$ by its column vectors

$$A_{(l)} = \begin{bmatrix} A_{(l),1} & A_{(l),2} & \cdots & A_{(l),M} \end{bmatrix} \tag{26}$$

where each column vector $A_{(l),i}$ can be conceptually viewed as a basis vector associated with the ith basis function. The inner product $< B_{(l)}, A_{(l),i} >$ measures the correlation between the measurement data $B_{(l)}$ and the basis vector $A_{(l),i}$. A strong correlation between $B_{(l)}$ and $A_{(l),i}$ implies that the basis vector $A_{(l),i}$ (hence, the ith basis function) is an important component to approximate $B_{(l)}$. Since we would like to identify a common set of basis functions for all wafers/dies, the following linear combination of inner products serves as a quantitative criterion for initial basis vector selection:

$$\underset{s}{\text{maximize}} \quad \sum_{l=1}^{L} \left| \left\langle B_{(l)}, A_{(l),s} \right\rangle \right|. \tag{27}$$

Equation (27) is expected to be more accurate than directly maximizing the inner product for any individual wafer/die, since it is less sensitive to the random noise caused by uncorrelated random variation and/or measurement error. In other words, by adding the inner products over L wafers/dies, the impact of random noise is reduced and the spatial pattern associated with systematic variation can be accurately detected.

We use the set Ω to denote the set of basis functions that have been selected. The set Ω consists of a single basis function after applying (27). Next, the coefficients associated with Ω are solved by least-squares fitting

$$\underset{\eta_{(l),i}, i \in \Omega}{\text{minimize}} \quad \left\| \sum_{i \in \Omega} A_{(l),i} \cdot \eta_{(l),i} - B_{(l)} \right\|_2^2 \quad (l = 1, 2, \ldots, L). \tag{28}$$

Solving (28) results in the residuals $\{e_{(l)}; l = 1, 2, \ldots, L\}$, which represent the spatial variation that cannot be represented by Ω

$$e_{(l)} = B_{(l)} - \sum_{i \in \Omega} A_{(l),i} \cdot \eta_{(l),i} \quad (l = 1, 2, \ldots, L). \tag{29}$$

In the next iteration, S-OMP further identifies the next important basis function by the largest total magnitude of inner product with the residual

$$\underset{s}{\text{maximize}} \quad \sum_{l=1}^{L} \left| \left\langle e_{(l)}, A_{(l),s} \right\rangle \right|. \tag{30}$$

Once the new basis function is selected, it is added to the set Ω and (28) is solved again to update the coefficients. If additional basis functions are needed, S-OMP will repeatedly select the optimal basis function according to (29) and then

Algorithm 1 Extended S-OMP

1. Start from the optimization problem in (23) with a given integer λ specifying the total number of basis vectors.
2. Initialize the set $\Omega = \Omega_0$, and the iteration index $p = 1$.
3. If $\Omega = \emptyset$, initialize the residuals $e_{(l)} = B_{(l)}$. Otherwise, initialize the residuals by applying (28)–(29).
4. Select the new basis vector s according to (30).
5. Update Ω by $\Omega = \Omega \cup \{s\}$.
6. Solve the least-squares fitting problems by (28).
7. Calculate the residuals $\{e_{(l)}; l = 1, 2, \ldots, L\}$ by (29).
8. If $p < \lambda$, $p = p + 1$ and go to Step 4.
9. For any $i \notin \Omega$, set $\eta_{(l),i} = 0$.

re-evaluate all coefficients by (28), until λ basis functions are selected in total.

Based on the aforementioned process, it is straightforward to extend the S-OMP algorithm when Ω_0 is nonempty. Suppose that Ω_0 contains λ_0 basis functions. We conceptually consider that λ_0 S-OMP iterations have been performed and the basis functions selected are in the set Ω_0. Therefore, we only need to resume S-OMP from this starting point. The flow of the extended algorithm is summarized in Algorithm 1. Note that Algorithm 1 relies on a user defined parameter λ to control the number of basis functions that should be selected. This parameter is estimated using cross-validation [25].

C. Robust S-OMP

In real-world measurement data, outliers typically exist because of manufacturing defects or measurement errors. For example, wafer probe test may produce incorrect measurement results due to probe misalignment [22]. Once outliers occur, they present themselves as abnormal data that significantly deviate from the regular range of parametric variation. A substantial error can be introduced by outliers if they are not appropriately detected and removed. Namely, sparse regression may not select the correct basis functions to model the spatially correlated variation. A few outliers can result in an extremely strong random variation component, thereby underestimating the spatially correlated component.

Traditionally, outlier detection can be performed by pre-processing all measurement data with the interquartile range (IQR) method [27]. The IQR is defined as

$$IQR = Q_3 - Q_1 \tag{31}$$

where $[Q_1 \;\; Q_2 \;\; Q_3]$ are the three values in ascending order which divide the sorted data into four equal parts. Next, for each measurement point, it is considered an outlier, if its value is outside the following variation range:

$$R_{IQR} = [Q_1 - 3 \cdot IQR,\, Q_3 + 3 \cdot IQR] \tag{32}$$

where the scaling factor 3 is decided empirically by the statistics community. If the measurement data is normally distributed, the IQR method removes the data outside the $\pm 4.7\sigma$ range.

In practice, we find the IQR method ineffective in detecting outliers for our application. The fundamental reason is that modern manufacturing processes suffer from a lot of variation sources. When directly viewing all measurement data, the accumulation of these variation sources will result in a very large variation range. Therefore, even if the outcome of a particular process step is strongly distorted by defects, such a distortion may not be significant compared to the natural variation range of all process steps, making outliers difficult to detect. Motivated by these observations, we propose a new outlier detection algorithm that aims to define the variation range based on uncorrelated random variation only. As such, the variation range for normal data can be significantly reduced, thereby making normal data and outliers easily separable. We introduce the proposed algorithm by first rewriting the matrix $A_{(l)}$ in (23) into a row matrix

$$A_{(l)} = \begin{bmatrix} A_{1,(l)} \\ A_{2,(l)} \\ \vdots \\ A_{N_{(l)},(l)} \end{bmatrix} \tag{33}$$

where each row corresponds to the value of basis functions at a particular measurement point. For each measurement point, we obtain the following residual after solving (23):

$$e_{(l),i} = b_{(l),i} \;\; - A_{i,(l)}\eta_{(l)} \quad \left(i = 1, 2, \ldots, N_{(l)}\right) \tag{34}$$

where $b_{(l),i}$ is the ith element of the vector $B_{(l)}$. Therefore, we can rewrite the sparse regression problem (23) as

$$\begin{array}{ll} \underset{\eta_{(l)}}{\text{minimize}} & \sum_{i=1}^{N_{(l)}} \rho_{L2}\left(e_{(l),i}\right) \\ \text{s.t.} & nnz\left(\left\{\eta_{(l),j} | j \notin \Omega_0\right\}\right) \le \lambda \end{array} \quad (l = 1, 2, \ldots, L) \tag{35}$$

where

$$\rho_{L2}(e) = e^2. \tag{36}$$

Based on the M-estimate theory in statistics [27], the regression problem (35) is sensitive to outliers, since the ρ function (36) is not robust. To understand this concept, we plot the function (36) in Fig. 4(a). It can be seen that as the residual moves away from zero, the objective function increases rapidly. Therefore, if large outliers exist, even if they are few in number, they can significantly influence the cost function (36) and strongly bias the result. For this reason, we adopt a robust error function called bisquare function [27], as shown in Fig. 4(b). Intuitively, since the value of a bisquare function stops growing after a certain threshold, it would prevent a small number of outliers from significantly biasing the result. Mathematically, the bisquare ρ function is defined as

$$\rho_{BS}(e) = \begin{cases} \dfrac{k^2}{6}\left(1 - \left(1 - \left(\dfrac{e}{k}\right)^2\right)^3\right) & (|e| \le k) \\ \dfrac{k^2}{6} & (|e| > k) \end{cases} \tag{37}$$

where k is a tuning constant specifying the cut-off threshold in Fig. 4(b). The following tuning constant is often used [27]:

$$k_{(l)} = 6.946 \cdot mad_i\left(e_{(l),i}\right) \tag{38}$$

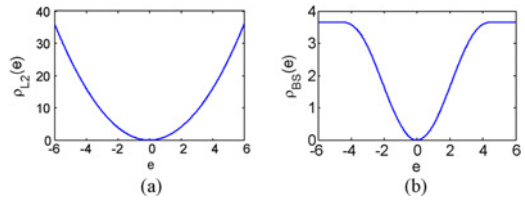


Fig. 4. (a) L_2-norm ρ function. (b) Bisquare ρ function.

Algorithm 2 Robust S-OMP

1. Apply Algorithm 1 to calculate the coefficients $\{\eta_{(l)}; l = 1, 2, \ldots, L\}$.
2. Calculate the weight for each measurement point according to the following weight function [27]:

$$w_{BS}\left(e_{(l),i}\right) = \begin{cases} 1 - \left(\frac{e_{(l),i}}{k_{(l)}}\right)^2 & \left(\left|e_{(l),i}\right| \le k_{(l)}\right) \\ 0 & \left(\left|e_{(l),i}\right| > k_{(l)}\right) \end{cases} \tag{41}$$

where $k_{(l)}$ is defined in (38)–(39).
3. Apply Algorithm 1 to solve the following problem:

$$\begin{array}{ll} \underset{\eta_{(l)}}{\text{minimize}} & \left\| W_{(l)}\left(A_{(l)}\eta_{(l)} - B_{(l)}\right)\right\|_2^2 \quad (l = 1, 2, \cdots, L) \\ s.t. & nnz\left(\left\{\eta_{(l),j} | j \notin \Omega_0\right\}\right) \le \lambda \end{array} \tag{42}$$

where $W_{(l)}$ is an diagonal matrix with $W_{(l),ii} = w_{BS}(e_{(l),i})$.
4. If the change of coefficients $\{\eta_{(l)}; l = 1, 2, \ldots, L\}$ is sufficiently small compared to the previous iteration step, stop. Otherwise go to Step 2.

where $mad_i(e_{(l),i})$ means the median absolute deviation of the residuals $\{e_{(l),i}; i = 1, 2, \ldots, N_{(l)}\}$

$$mad_i\left(e_{(l),i}\right) = median_i\left(\left|e_{(l),i} - median_j\left(e_{(l),j}\right)\right|\right)/0.6745. \tag{39}$$

When the residual is normally distributed, the cut-off threshold $k_{(l)}$ corresponds to 4.685σ variation, which is similar to that in (32) based on 3·IQR.

We formulate the robust sparse regression problem as

$$\begin{array}{ll} \underset{\eta_{(l)}}{\text{minimize}} & \sum_{i=1}^{N_{(l)}} \rho_{BS}\left(e_{(l),i}\right) \\ \text{s.t.} & nnz\left(\left\{\eta_{(l),j} | j \notin \Omega_0\right\}\right) \le \lambda \end{array} \quad (l = 1, 2, \ldots, L) \tag{40}$$

where the definition of ρ_{BS} is in (37)–(39). We borrow the iteratively reweighted least-squares method [27] from statistics to solve this problem. With the solution of (35) as a starting point, the algorithm first calculates the weight for each measurement point at each iteration step, according to a weight function derived from the error function (37). Next, it solves an optimization problem with a weighted L_2-norm cost function. These steps are summarized in Algorithm 2.

Since Algorithm 1 is applied to solve a sparse regression problem in each iteration step, its computational cost is approximately equal to the runtime of Algorithm 1 multiplied by the number of iterations. We observe that Algorithm 2 will converge within ten iterations in most cases, even though the global convergence is not proven. Therefore, the computational time increases by about 2–10× compared to Algorithm 1. After performing Algorithm 2, the measurement points for which the residual exceeds the cutoff threshold in (38) will have zero weight in (41). These points are identified as outliers and removed before solving the mixed model (4).

V. Implementation Details

As discussed in the previous section, the computational cost of Algorithm 2 is proportional to that of performing Algorithm 1. The computational cost of Algorithm 1 depends on the size of the dictionary, which is typically small when the physical dictionary is applied only. However, when the DCT dictionary is applied, the total number of basis functions can be extremely large for wafers with small die size or test chips with a large number of test structures. Studying Algorithm 1, we observe that its computational cost is dominated by two steps: the inner product computation in Step 4 and the least-squares fitting in Step 6. We will discuss the numerical algorithm to speed up these two steps in this section.

A. Inner Product Computation

When applying Algorithm 1 to solve (42) with physical and DCT dictionaries to select the basis vectors using (30), we will need to compute the following inner product values

$$\left\langle e_{(l)}, A_{w(l),j}\right\rangle \quad (j = 1, 2, \cdots, (\lambda_0 + PQ), \quad l = 1, 2, \cdots, L) \tag{43}$$

where λ_0 is the number of selected physical basis functions, PQ is the number of DCT basis functions, and

$$A_{w(l)} = W_{(l)} \cdot A_{(l)}. \tag{44}$$

A straightforward implementation first directly computes $A_{w(l)}$ by (44), and then calculates (43) by vector–vector multiplications. The computational cost is in the order of $O(LPQ(\lambda_0 + PQ))$. Note that the computational cost quadratically increases with the DCT dictionary size PQ. Hence, this implementation can quickly become computationally intractable, as the problem size increases. To derive an efficient algorithm, we first rewrite (43) as

$$\left\langle e_{(l)}, A_{w(l),j}\right\rangle = A_{w(l),j}^T \cdot e_{(l)}. \tag{45}$$

For each $l \in \{1, 2, \ldots, L\}$, we need to calculate (43) for each basis vector, i.e., $j \in \{1, 2, \ldots, \lambda_0 + PQ\}$. The results can be expressed by the following matrix–vector multiplications:

$$\begin{bmatrix} \left\langle e_{(l)}, A_{w(l),1}\right\rangle \\ \left\langle e_{(l)}, A_{w(l),2}\right\rangle \\ \vdots \\ \left\langle e_{(l)}, A_{w(l),M}\right\rangle \end{bmatrix} = A_{w(l)}^T \cdot e_{(l)} = A_{(l)}^T \cdot W_{(l)} \cdot e_{(l)}. \tag{46}$$

In other words, by calculating the matrix–vector multiplications in (46), we are able to obtain the inner product values for all $\lambda_0 + PQ$ basis vectors. Since $W_{(l)}$ is a diagonal matrix, the matrix–vector multiplication $W_{(l)} \cdot e_{(l)}$ can be first simply performed with linear complexity

$$e_{w(l)} = W_{(l)} \cdot e_{(l)}. \tag{47}$$

Next, we need to efficiently compute the matrix–vector product $A_{(l)}^T \cdot e_{w(l)}$, which can be rewritten into the following form:

$$A_{(l)}^T \cdot e_{w(l)} = \begin{bmatrix} A_{\Omega_0(l)}^T \cdot e_{w(l)} \\ A_{dct(l)}^T \cdot e_{w(l)} \end{bmatrix} \tag{48}$$

where $A_{\Omega_{(l)}}$ contains the columns of $A_{(l)}$ that correspond to the λ_0 preselected physical basis functions, and $A_{dct(l)}$ contains the columns of $A_{(l)}$ that correspond to all PQ DCT basis functions. Since the number of all DCT basis functions should be much larger than the number of preselected physical basis functions, the key bottleneck for computing (48) is the second term $A^T_{dct(l)} \cdot e_{w(l)}$. We observe that if the measurement of the lth wafer/die does not contain any missing data, the matrix $A_{dct(l)}$ represents the IDCT matrix and it is a full-rank square matrix. In this case, since DCT/IDCT is an orthogonal transform [26], $A^T_{dct(l)} = A^{-1}_{dct(l)}$ is exactly the DCT matrix. Namely, calculating the matrix–vector product $A^T_{dct(l)} \cdot e_{w(l)}$ is equivalent to performing DCT on $e_{w(l)}$. Similar to fast Fourier transform (FFT), there exist a number of fast algorithms for DCT/IDCT. The computational cost of these fast algorithms is in the order of $O(PQ \cdot log(PQ))$ [26]. Therefore, by using a fast DCT algorithm, the computational cost for Step 4 of Algorithm 1 is reduced from $O(LPQ(\lambda_0 + PQ))$ to $O(LPQ(\lambda_0 + \log(PQ)))$. It, in turn, brings significant speedup, since λ_0, should be much smaller than PQ.

The aforementioned fast DCT algorithm is applicable only if there is no missing data. If a number of missing data exist (e.g., due to measurement error), we can construct an augmented vector $e^*_{w(l)} \in R^{PQ}$ where its elements corresponding to missing data are simply filled with zeros. Mathematically, the augmented vector $e^*_{w(l)}$ can be represented as

$$e^*_{w(l)} = Z_{(l)} \cdot \begin{bmatrix} e_{w(l)} \\ 0 \end{bmatrix} \tag{49}$$

where $Z_{(l)}$ is a permutation matrix to map $e_{w(l)}$ and the zero vector to the appropriate elements in $e^*_{w(l)}$. Applying DCT to the augmented vector $e^*_{w(l)}$ yields

$$A^{*T} \cdot e^*_{w(l)} = A^{*T} \cdot Z_{(l)} \cdot \begin{bmatrix} e_{w(l)} \\ 0 \end{bmatrix} \tag{50}$$

where A^* represents the IDCT matrix and, hence, $A^{*\mathrm{T}}$ is the DCT matrix. Remember that the matrix $A_{dct(l)}$ contains $N_{(l)}$ rows taken from the IDCT matrix A^*. Hence, if $Z_{(l)}$ is appropriately chosen, the matrix $A^{*\mathrm{T}} \cdot Z_{(l)}$ in (50) can be rewritten as

$$A^{*T} \cdot Z_{(l)} = \begin{bmatrix} A^T_{dct(l)} & A^T_{dct(\bar{l})} \end{bmatrix} \tag{51}$$

where the matrix $A_{dct(\bar{l})}$ contains the $PQ - N_{(l)}$ rows of A^* that are not included in $A_{dct(l)}$ due to missing data. Substituting (51) into (50), we have

$$A^{*T} \cdot e^*_{w(l)} = \begin{bmatrix} A^T_{dct(l)} & A^T_{dct(\bar{l})} \end{bmatrix} \cdot \begin{bmatrix} e_{w(l)} \\ 0 \end{bmatrix} = A^T_{dct(l)} \cdot e_{w(l)}. \tag{52}$$

Note that the DCT results in (52) are exactly equal to the second matrix–vector product in (48). It, in turn, demonstrates that by filling the missing data with zeros, we can efficiently calculate the inner product values by using a fast DCT algorithm. In this case, the computational cost for Step 4 of Algorithm 1 is again reduced from $O(LPQ(\lambda_0 + PQ))$ to $O(LPQ(\lambda_0 + \log(PQ)))$.

In addition to the reduction in computational cost, the fast algorithm based on DCT can also efficiently reduce the memory consumption. Note that the direct matrix–vector multiplication in (46) requires explicitly forming a dense matrix $A_{(l)}$ with about $PQ(\lambda_0 + PQ)$ entries. While it is possible to calculate each inner product in (46) one by one without forming the matrix $A_{(l)}$, such an approach leads to large computational time since each column of $A_{(l)}$ must be repeatedly formed during the iterations of Algorithm 1. For these reasons, the direct approach based on matrix–vector multiplication or vector–vector multiplication is expensive in either memory consumption or computational time. On the other hand, our proposed method only needs to form the submatrix $A_{\Omega_0(l)}$ with $PQ \cdot \lambda_0$ entries. A fast DCT algorithm can be applied to $e^*_{w(l)}$ without explicitly building the DCT matrix $A_{dct(l)}$ in memory, thereby significantly reducing the memory consumption for large-scale problems.

B. Least-Squares Fitting

In addition to inner product computation, least-squares fitting is another computationally expensive operation that is required by Step 6 of Algorithm 1. For the lth wafer/die at the pth iteration step, the following optimization problem needs to be solved:

$$\underset{\eta_{(l),(p)}}{\text{minimize}} \quad \left\| W_{(l)} \cdot A_{(l),(p)} \cdot \eta_{(l),(p)} - B_{(l)} \right\|_2^2 \tag{53}$$

where the matrix $A_{(l),(p)}$ contains $\lambda_0 + p$ column vectors selected from $A_{(l)}$ and the vector $\eta_{(l),(p)}$ contains the coefficients corresponding to these selected basis vectors. Similarly, we can rewrite $A_{(l),(p)}$ into the following:

$$A_{(l),(p)} = \begin{bmatrix} A_{\Omega_0(l)} & A_{dct(l),(p)} \end{bmatrix} \tag{54}$$

where the matrix $A_{dct(l),(p)}$ contains p column vectors selected from $A_{dct(l)}$. The relation between $A_{dct(l),(p)}$ and $A_{dct(l)}$ can be further expressed as

$$A_{dct(l)} \cdot Z_{(p)} = \begin{bmatrix} A_{dct(l),(p)} & A_{dct(l),(\bar{p})} \end{bmatrix} \tag{55}$$

where $Z_{(p)}$ is a permutation matrix, and the matrix $A_{dct(l),(\bar{p})}$ contains the DCT basis vectors that are not included in $A_{dct(l),(p)}$.

The least-squares solution $\eta_{(l),(p)}$ of (53) satisfies the following normal equation [28]:

$$\left(W_{(l)} \cdot A_{(l),(p)}\right)^T \cdot \left(W_{(l)} \cdot A_{(l),(p)}\right) \cdot \eta_{(l),(p)} = \left(W_{(l)} \cdot A_{(l),(p)}\right)^T \cdot B_{w(l)}. \tag{56}$$

Traditionally, the solution $\eta_{(l),(p)}$ of (56) is solved by QR decomposition [28]. The computational cost is in the order of $O(N_{(l)} \cdot (\lambda_0 + p)^2)$, which is prohibitively expensive for large-scale problems. An alternative way to solve (56) is based on an iterative algorithm that is referred to as the LSQR method [12]. LSQR relies on the bi-diagonalization of the matrix $A_{(l),(p)}$. During its iterations, LSQR generates a sequence of solutions to approximate $\eta_{(l),(p)}$. The solutions are exactly identical to the results calculated by the conjugate gradient method [28] for the normal equation (56), but LSQR achieves better numerical stability than the conjugate gradient method. The details of LSQR can be found in [12].

When applying LSQR, it is not necessary to explicitly form the matrix $A_{(l),(p)}$. Instead, in each iteration, only two operations need to be performed, $W_{(l)} \cdot A_{(l),(p)} \cdot \alpha$ and $A^T_{(l),(p)} \cdot W_{(l)} \cdot \beta$,

where α is a $(\lambda_0 + p)$-by-1 vector and β is an $N_{(l)}$-by-1 vector. Similar to Section V-A, these matrix–vector multiplications can be efficiently calculated by applying a fast numerical algorithm based on fast DCT/IDCT transform.

To efficiently compute $W_{(l)} \cdot A_{(l),(p)} \cdot \alpha$, we only need to efficiently compute $A_{(l),(p)} \cdot \alpha$, and the multiplication with the matrix $W_{(l)}$ is a simple vector–vector product, similarly to (47). Therefore, we rewrite $A_{(l),(p)} \cdot \alpha$ into the following equation:

$$A_{(l),(p)} \cdot \alpha = A_{\Omega_0(l)} \cdot \alpha_{\Omega_0} + A_{dct(l),(p)} \cdot \alpha_{dct} \tag{57}$$

where $\alpha_{\Omega 0}$ is an λ_0-by-1 vector and α_{dct} is a p-by-1 vector. We are able to efficiently compute $A_{dct(l),(p)} \cdot \alpha_{dct}$ by constructing an augmented vector $\alpha^*_{dct} \in R^{PQ}$

$$\alpha^*_{dct} = Z_{(p)} \cdot \begin{bmatrix} \alpha_{dct} \\ 0 \end{bmatrix} \tag{58}$$

where $Z_{(p)}$ is the permutation matrix defined in (55). If we conceptually consider α_{dct} as a vector of selected DCT coefficients, α^*_{dct} represents all DCT coefficients with the unselected DCT coefficients filled by 0. We then apply IDCT to the augmented vector α^*_{dct}

$$A^* \cdot \alpha^*_{dct} = A^* \cdot Z_{(p)} \cdot \begin{bmatrix} \alpha_{dct} \\ 0 \end{bmatrix} \tag{59}$$

where A^* denotes the IDCT matrix as defined in (50). On the other hand, we can derive the following equation from (51):

$$A^* = Z_{(l)} \cdot \begin{bmatrix} A_{dct(l)} \\ A_{dct(\bar{l})} \end{bmatrix}. \tag{60}$$

Substituting (60) for (59) yields

$$A^* \cdot \alpha^*_{dct} = Z_{(l)} \cdot \begin{bmatrix} A_{dct(l)} \cdot Z_{(p)} \\ A_{dct(\bar{l})} \cdot Z_{(p)} \end{bmatrix} \cdot \begin{bmatrix} \alpha_{dct} \\ 0 \end{bmatrix}. \tag{61}$$

In (61), $A_{dct(l)}.Z_{(p)}$ can be represented as two submatrices as shown in (55). If we similarly rewrite $A_{dct(l)}.Z_{(p)}$ as two submatrices

$$A_{dct(\bar{l})} \cdot Z_{(p)} = \begin{bmatrix} A_{dct(\bar{l}),(p)} & A_{dct(\bar{l}),(\tilde{p})} \end{bmatrix}. \tag{62}$$

Equation (61) becomes

$$Z_{(l)} \cdot \begin{bmatrix} A_{dct(l),(p)} & A_{dct(l),(\tilde{p})} \\ A_{dct(\bar{l}),(p)} & A_{dct(\bar{l}),(\tilde{p})} \end{bmatrix} \cdot \begin{bmatrix} \alpha_{dct} \\ 0 \end{bmatrix} = Z_{(l)} \cdot \begin{bmatrix} A_{dct(l),(p)} \cdot \alpha_{dct} \\ A_{dct(\bar{l}),(p)} \cdot \alpha_{dct} \end{bmatrix}. \tag{63}$$

Since $Z_{(l)}$ is a permutation matrix, (63) is equivalent to

$$\begin{bmatrix} A_{dct(l),(p)} \cdot \alpha_{dct} \\ A_{dct(\bar{l}),(p)} \cdot \alpha_{dct} \end{bmatrix} = Z^T_{(l)} \cdot A^* \cdot \alpha^*_{dct}. \tag{64}$$

Equation (64) reveals an important fact that the matrix–vector multiplication $A_{dct(l),(p)} \cdot \alpha_{dct}$ can be efficiently computed by applying IDCT to the augmented vector α^*_{dct}. The value of $A_{dct(l),(p)} \cdot \alpha_{dct}$ is determined by selecting the appropriate elements from the IDCT result $A^* \cdot \alpha^*{}_{dct}$. If a fast IDCT algorithm is applied [26], the computational cost of this matrix–vector multiplication is in the order of $O(PQ \cdot \log(PQ))$. Therefore, the computational cost of the operation $W_{(l)} \cdot A_{(l),(p)} \cdot \alpha$ is $O(PQ \cdot (\lambda_0 + \log(PQ)))$.

Next, we consider the other matrix–vector multiplication $A^T_{(l),(p)} \cdot W_{(l)} \cdot \beta$ that is required by the LSQR algorithm. Similarly to the computation in (46)–(47), $W_{(l)} \cdot \beta$ can be first computed with linear complexity

$$\beta_w = W_{(l)} \cdot \beta. \tag{65}$$

Next, we rewrite $A_{T(l),(p)} \cdot \beta_w$ into the following equation:

$$A^T_{(l),(p)} \cdot \beta_w = \begin{bmatrix} A^T_{\Omega_0(l)} \cdot \beta_w \\ A^T_{dct(l),(p)} \cdot \beta_w \end{bmatrix}. \tag{66}$$

We are able to efficiently compute $A^T_{dct(l),(p)} \cdot \beta_w$ by first constructing an augmented vector $\beta^*_w \in R^{PQ}$

$$\beta^*_w = Z_{(l)} \cdot \begin{bmatrix} \beta_w \\ 0 \end{bmatrix} \tag{67}$$

where $Z_{(l)}$ is the permutation matrix defined in (49). Similarly to (49), if we conceptually consider β_w as a vector of available measurements, $\beta^*{}_w$ represents all measurements with the missing data filled by 0. We then apply DCT to the augmented vector β^*_w

$$A^{*T} \cdot \beta^*_w = A^{*T} \cdot Z_{(l)} \cdot \begin{bmatrix} \beta_w \\ 0 \end{bmatrix} \tag{68}$$

where $A^{*\mathrm{T}}$ is the DCT matrix as defined in (50). Substituting (60) for (68) yields

$$A^{*T} \cdot \beta^*_w = \begin{bmatrix} A^T_{dct(l)} & A^T_{dct(\bar{l})} \end{bmatrix} \cdot Z^T_{(l)} \cdot Z_{(l)} \cdot \begin{bmatrix} \beta_w \\ 0 \end{bmatrix} = A^T_{dct(l)} \cdot \beta_w. \tag{69}$$

Based on (55), (69) can be further rewritten as

$$\begin{bmatrix} A^T_{dct(l),(p)} \cdot \beta_w \\ A^T_{dct(l),(\tilde{p})} \cdot \beta_w \end{bmatrix} = Z^T_{(p)} \cdot A^{*T} \cdot \beta^*_w. \tag{70}$$

Hence, the matrix–vector multiplication $A^T_{dct(l),(p)} \cdot \beta_w$ can be calculated by applying DCT to the augmented vector β^*_w. The value of $A^T_{dct(l),(p)} \cdot \beta_w$ is determined by selecting the appropriate elements from the DCT result $A^{*T} \cdot \beta^*_w$. The computational cost is in the order of $O(PQ \cdot \log(PQ))$. Therefore, the computational cost of the operation $A^T_{(l),(p)} \cdot W_{(l)} \cdot \beta$ is also $O(PQ \cdot (\lambda_0 + \log(PQ)))$.

Finally, it is worth mentioning that similar to other iterative solvers, a good initial guess should be provided to LSQR to achieve fast convergence. If the initial guess is close to the actual solution, LSQR can reach convergence in a few iterations [12]. In this paper, LSQR is required at each iteration step of Algorithm 1. For each iteration step, the solution from the previous iteration step can serve as a good initial guess for the current iteration step. By adopting such a heuristic, LSQR typically converges in only two to three iterations in our tested examples.

It should be noted that the aforementioned fast implementation is mainly based on fast matrix–vector product, which is an elementary operation in many algorithms. Therefore, it is possible to apply the same idea to speed up other sparse regression algorithms, such as group lasso [11].

VI. Numerical Experiments

In this section, we demonstrate the accuracy and computational efficiency of our proposed variation decomposition

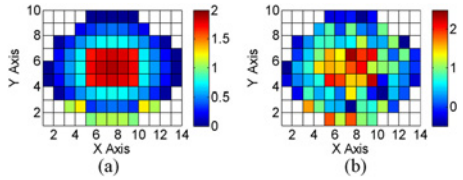


Fig. 5. (a) Systematic variation of the synthetic wafer. (b) Synthetic data created by adding extra random variation.

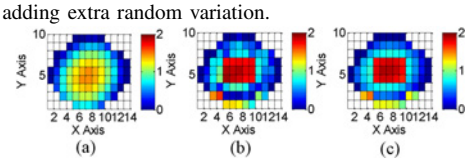


Fig. 6. (a) Spatially correlated variation extracted by quadratic basis functions only. (b) Spatially correlated variation extracted by the physical dictionary without sparse regression. (c) Spatially correlated variation extracted by the proposed method with the physical dictionary.

algorithm using several examples. All numerical experiments are performed on a 2.8 GHz Linux server.

A. Synthetic Data

We first consider a synthetic example where the systematic variation contains quadratic, edge, and center effects created with five basis functions. The quadratic pattern is created by

$$s(x, y) = 1 - x^2 - y^2 \tag{71}$$

where x and y are the coordinates on the wafer with range normalized to $[-1\ 1]$. Equation (71) creates a decreasing radial pattern. The edge effect only occurs at the bottom edge of the wafer, and the center effect is created by the basis function in Fig. 3(b). The magnitude of the edge and center effects is adjusted so that each effect contributes to one third of the variance in systematic variation. The systematic wafer map is shown in Fig. 5(a). The synthetic data is created by adding extra random variation distributed as $N(0, 0.4^2)$, which is shown in Fig. 5(b). After adding the random variation, the systematic variation contributes to 65.9% of the total variance.

We first apply REML with only the quadratic basis functions in (7). Fig. 6(a) shows the extracted spatially correlated variation. It can be intuitively seen that the edge and center effects are not adequately captured. The estimated spatially correlated variation is 43.2%, which underestimates the true systematic variation by a large amount. Next, we apply the proposed physical dictionary, which contains the quadratic basis functions in (7), all the depth 1 and 2 edge basis functions with the different partitions in Fig. 2, and the center basis functions in Fig. 3. Fig. 6(b) shows the spatially correlated variation extracted by REML with all basis functions from the physical dictionary. The estimated spatially correlated variation is 74.7%. It can be seen that while the results are more accurate than Fig. 6(a), it overestimates the spatially correlated variation due to overfitting. Fig. 6(c) shows the spatially correlated variation extracted by the proposed method with the physical dictionary, which applies REML to the basis functions selected by Algorithm 2 only. The proposed method identifies 11 basis functions from the dictionary, and the estimated spatially correlated variation is 70.1%. Although the proposed method does not completely remove overfitting, the spatial pattern is intuitively more accurate than Fig. 6(b), and the estimation is much closer to the actual percentage. This, in turn, demonstrates that the proposed sparse regression method significantly reduces the inaccuracy caused by overfitting.

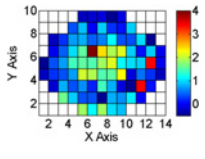


Fig. 7. Synthetic data after adding three outliers.

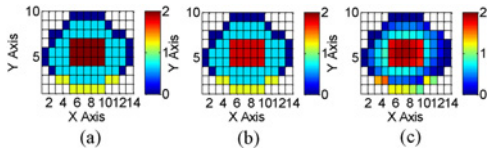


Fig. 8. (a) Spatially correlated variation extracted by sparse regression using the physical dictionary without outlier detection. (b) Spatially correlated variation extracted by sparse regression using the physical dictionary with traditional outlier detection. (c) Spatially correlated variation extracted by the proposed method based on robust sparse regression using the physical dictionary.

We further examine the efficacy of the proposed outlier detection technique by randomly adding three outliers at three random locations in Fig. 5(b). The resulting wafer map is shown in Fig. 7. For each location, the outlier is created by adding 3·IQR of the wafer to its original value, where IQR is defined in (31).

Fig. 8(a) shows the extracted spatially correlated variation with 6 basis functions by directly applying Algorithm 1. The estimated spatially correlated variation is 48.3%, which significantly underestimates the spatially correlated variation. Examining Fig. 8(a), it can be seen that it does not contain the radial pattern produced by the quadratic basis functions in Fig. 5(a). Next, the traditional outlier detection method detects only the outlier located in the center of the wafer, and the extracted spatially correlated variation is shown in Fig. 8(b). The estimated spatially correlated variation is 49.0%. Examining Fig. 8(b), it can be seen that the same basis functions are selected and therefore it fails to capture the radial pattern. In other words, no significant improvement has been achieved. Finally, the robust sparse regression method correctly detects all three outliers. The extracted spatially correlated variation is shown in Fig. 8(c). The estimated spatially correlated variation is 71.5%. Compared to Fig. 6(c), the same basis functions are selected with no significant accuracy loss.

Besides the variation sources modeled by the physical dictionary, there exist other sources that are not well understood. In this case, the DCT dictionary can be applied to discover any significant spatial pattern that has been missed by the physical dictionary. For wafer-level variation, one possible scenario is the spatial variation caused by multiple heat sources [19] or a heat source with complicated shape [20]. For within-die variation, an example is the mask error. One possible outcome of mask error is that a significant mean shift may exist between two portions of a die. We construct a systematic within-die variation map in Fig. 9(a), which has a significant mean shift at x = 8. The synthetic data is created by adding extra random variation distributed as N(0, 0.32), which is shown in Fig. 9(b).

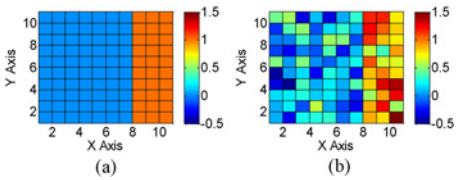


Fig. 9. (a) Systematic variation of the synthetic die. (b) Synthetic data created by adding extra random variation.

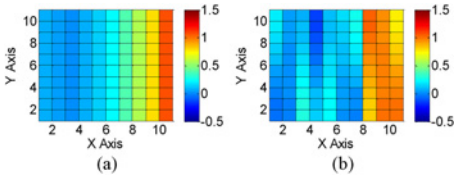


Fig. 10. (a) Spatially correlated variation extracted by the proposed method with the physical dictionary. (b) Spatially correlated variation extracted by the proposed method with the physical and DCT dictionaries.

After adding the random variation, the systematic variation contributes to 73.1% of the total variance.

Fig. 10(a) shows the spatially correlated variation extracted by the proposed method with the physical dictionary, which contains quadratic basis functions. In this example, three basis functions are identified, and the estimated spatially correlated variation is only 54.1%. Fig. 10(b) shows the spatially correlated variation extracted by the proposed method with the physical and DCT dictionaries. Here, six DCT basis functions are selected, and the estimated spatially correlated variation is 70.7%. Comparing Fig. 10(b) with Fig. 10(a), it can be seen that the mean shift is clearly revealed, which serves as a good basis for helping the process engineers to identify the source of variation.

B. Silicon Measurement Data

From the previous experiments, we observe that by applying robust sparse regression, we are able to accurately find the basis functions and detect the outliers for the synthetic data. This subsection presents the results of performing variation decomposition on several data sets of silicon measurements.

We first consider the transistor drain saturation current (I_{dsat}) measurements taken from the scribe line test structures from eight wafers of a commercial CMOS process below 90 nm. Fig. 11 shows one of the representative wafers. Intuitively, the measurement data contain significant random variation. Table I compares the variation components estimated by three methods: (i) REML with quadratic basis functions, (ii) the proposed method with the physical dictionary, and (iii) the proposed method with the physical and DCT dictionaries. Fig. 12 compares the spatially correlated variation extracted by these three methods. From Table I, it can be seen that compared to REML using the quadratic basis functions only, robust sparse regression with the physical dictionary explains a significantly larger amount of variation as wafer-level spatially correlated variation. From Fig. 12(b), it can be intuitively seen that more obvious edge and center effect patterns are modeled than Fig. 12(a). Therefore, it reveals that edge and center effects contribute significantly to the wafer-level variation in this example. After adding the DCT basis functions, we do not observe significant increase in wafer-level spatially correlated

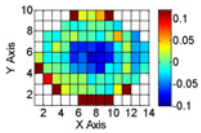


Fig. 11. I_{dsat} measurement data (normalized) from one of the eight wafers.

TABLE I

VARIATION COMPONENTS OF I_{DSAT} MEASUREMENT DATA

Method	Wafer-to -Wafer	Wafer-Level Spatially Correlated	Wafer-Level Random
Quadratic	30.2%	45.2%	24.6%
Proposed physical	28.2%	53.8%	17.9%
Proposed physical+DCT	30.3%	54.8%	14.9%

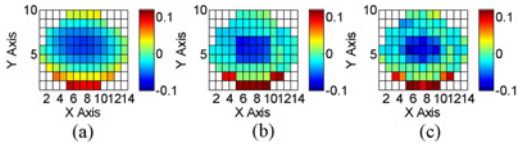


Fig. 12. (a) Spatially correlated variation extracted by directly applying quadratic basis functions. (b) Spatially correlated variation extracted by the proposed method with the physical dictionary. (c) Spatially correlated variation extracted by the proposed method with the physical and DCT dictionaries.

variation, and Fig. 12(c) does not clearly show any additional meaningful pattern compared to Fig. 12(b). Therefore, we believe that the physical dictionary is sufficient in modeling the spatially correlated variation in this example.

Next, we consider the contact plug resistance measurement data collected from 24 test chips in a 90 nm CMOS process. Each chip contains 36,864 test structures (i.e., contacts) arranged as a 144 × 256 array [23]. Contacts with 55 different layout patterns are regularly distributed over the entire chip. The spatial distribution of different layout patterns is shown in Fig. 13(a). Fig. 13(b) shows the measured contact plug resistance (normalized) from one of the 24 test chips. Studying Fig. 13(b), we would notice that there is a unique spatial pattern due to layout dependency. However, the spatial pattern is not clearly visible because of the large-scale uncorrelated random variation in this example.

We first extract the spatially correlated variation by robust sparse regression with the physical dictionary. In addition to the quadratic basis functions, since we know that the different layout patterns must be an important component of spatial variation, we construct 55 indicator basis functions according to (10) in the physical dictionary and preselect them in the sparse regression process. The extracted spatially correlated variation is shown in Fig. 14(a). It closely matches the layout pattern distribution in Fig. 13(a), which shows that the layout-dependent variation is the dominant variation source. To examine whether there exists any significant variation source that is not captured, we further perform sparse regression after adding the DCT dictionary. To avoid high computational cost, we simply apply Algorithm 1 after removing the outliers detected in the previous step. Fig. 14(b) shows the spatially correlated variation extracted by sparse regression with the physical and DCT dictionaries. From Table II, it can be seen that the variation percentages are not significantly different from the previous experiment, meaning that there do not exist additional significant variation sources. However, comparing Fig. 14(b) with Fig. 14(a), we notice that there is a subtle left-to-right

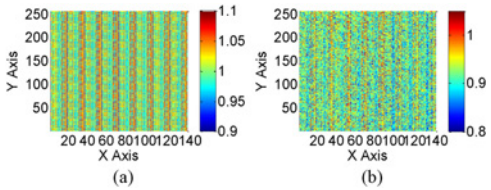


Fig. 13. (a) Spatial distribution of different contact layout patterns in the test chip. (b) Measured contact resistance (normalized) from one of the test chips.

TABLE II
VARIATION COMPONENTS OF CONTACT RESISTANCE MEASUREMENT DATA

Method	Wafer-Level	Within-die Spatially Correlated	Within-Die Random
Proposed physical	51.5%	30.9%	17.6%
Proposed physical+DCT	51.5%	31.5%	17.0%

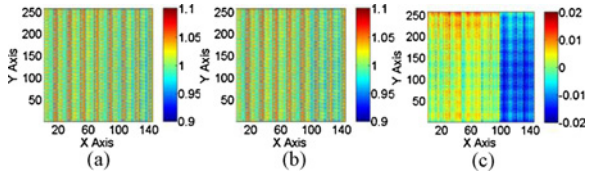


Fig. 14. (a) Spatially correlated variation extracted by the physical dictionary. (b) Spatially correlated variation extracted by the physical and DCT dictionaries. (c) Spatially correlated variation represented by the quadratic and DCT components.

transition at around $x = 100$. This transition becomes obvious if we plot only the quadratic and DCT components in Fig. 14(c). This sharp transition may be caused by mask error, which is a common source for within-die variation. Although this component is not significant in this example, it demonstrates that this type of variation can be revealed by applying our proposed sparse regression algorithm to the DCT dictionary.

In this example, since each test chip contains 36 864 test structures, the numerical algorithms developed in Section V become extremely critical. To demonstrate the efficiency of the proposed fast numerical algorithms, we implement three versions of Algorithm 1 where the inner product and the least-squares fitting are calculated by different methods. In the first implementation, the inner product is directly computed by (43) and the least-squares fitting is directly computed by QR decomposition. In the second implementation, the traditional inner product calculation is replaced by the fast algorithm proposed in Section V-A. Finally, in the third implementation, both the inner product and the least-squares fitting are calculated by the fast algorithms proposed in Section V. For testing and comparison purposes, we first run Algorithm 1 with the aforementioned three implementations for a single die in Fig. 13(a), and Table III shows the computational time. Note that the fast algorithm for inner product computation achieves 91× speedup and the fast least-squares fitting further brings 2.2× speedup. The overall speedup achieved by our proposed fast algorithms is 199×, compared to the traditional direct implementation.

Next, we apply Algorithm 1 to all 24 test chips, and Table IV compares the computational time for two different implementations. Once Algorithm 1 is applied to all test chips, the computational time increases significantly. The simple implementation with direct inner product calculation and least-squares fitting is not computationally feasible. Hence, its result is not shown in Table IV. In this example, the proposed fast algorithm for least-squares fitting achieves 2.1× speed-up over the direct implementation. This observation is consistent with the speedup in Table III. In addition, we infer that if the simple implementation with direct inner product calculation and least-squares fitting is adopted in this example, it would take more than one year to obtain the results, which makes Algorithm 1 inapplicable. Therefore, by applying the proposed fast implementation, we are able to extend Algorithm 1 to large problems.

TABLE III
COMPUTATIONAL TIME OF SPARSE REGRESSION FOR A SINGLE CHIP

Inner Product	Least-Squares Fitting	CPU Time (Hours)
Direct	Direct	514.9
Fast	Direct	5.6
Fast	Fast	2.6

TABLE IV
COMPUTATIONAL TIME OF SPARSE REGRESSION FOR 24 CHIPS

Inner product	Least-squares fitting	CPU time (Hours)
Fast	Direct	135.8
Fast	Fast	65.3

VII. CONCLUSION

In this paper, we proposed a new technique to achieve accurate decomposition of process variation by performing efficient spatial pattern analysis. The proposed technique applied sparse regression to accurately extract the most adequate basis functions to represent spatially correlated variation. Moreover, a robust sparse regression algorithm was proposed to automatically remove measurement outliers, and fast numerical algorithms were developed to reduce the computational time by several orders of magnitude over the traditional direct implementation. The effectiveness of the proposed technique was demonstrated by experimental results based on both synthetic and silicon data. For future research, we plan to further expand the physical dictionary to model additional physical effects, and extend the proposed approach to model wafer-to-wafer variation.

REFERENCES

[1] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.
[2] A. Gattiker, "Unraveling variability for process/product improvement," in *Proc. Int. Test Conf.*, 2008, pp. 1–9.
[3] S. Reda and S. Nassif, "Accurate spatial estimation and decomposition techniques for variability characterization," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 3, pp. 345–357, Aug. 2010.
[4] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-field gate length spatial variation for process-design co-optimization," in *Proc SPIE*, vol. 5756. May 2005, pp. 178–188.
[5] L. Cheng, P. Gupta, C. Spanos, K. Qian, and L. He, "Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability," *IEEE Trans. Comput.-Aided Des.*, vol. 30, no. 3, pp. 388–401, Mar. 2011.
[6] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," *IEEE Trans. Comput.-Aided Design*, vol. 26, no. 4, pp. 619–631, Apr. 2007.
[7] X. Li, R. Rutenbar, and R. Blanton, "Virtual probe: A statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," in *Proc. Int. Conf. Comput.-Aided Des.*, 2009, pp. 433–440.
[8] W. Zhang, X. Li, and R. Rutenbar, "Bayesian virtual probe: Minimizing variation characterization cost for nanoscale IC technologies via Bayesian inference," in *Proc. Des. Autom. Conf.*, 2010, pp. 262–267.

[9] 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," in *Proc. Int. Conf. Comput.-Aided Des.*, 2010, pp. 47–54.

[10] J. Tropp, A. Gilbert, and M. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, pp. 572–588, Mar. 2006.

[11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc. B*, vol. 68, no. 1, pp. 49–67, Feb. 2007.

[12] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982.

[13] V. Vahedi, M. Srinivasan, and A. Bailey, "Raising the bar on wafer edge yield: An etch perspective," *Solid State Technol.*, vol. 55, no. 11, Nov. 2008.

[14] L. Pang, K. Qian, C. Spanos, and B. Nikolic, "Measurement and analysis of variability in 45 nm strained-Si CMOS technology," *IEEE J. Solid-State Circuits*, vol. 44, no. 8, pp. 2233–2243, Aug. 2009.

[15] S. Sakai, M. Ogino, R. Shimizu, and Y. Shimogaki, "Deposition uniformity control in a commercial scale HTO-CVD reactor," in *Proc. Mater. Res. Soc. Symp.*, 2007.

[16] J. Brcka and R. L. Robison, "Wafer redeposition impact on etch rate uniformity in IPVD system," *IEEE Trans. Plasma Sci.*, vol. 35, no. 1, pp. 74–82, Feb. 2007.

[17] C. Chao, S. Hung, and C. Yu, "Thermal stress analysis for rapid thermal processor," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 2, pp. 335–341, May 2003.

[18] J. Hebb and K. Jensen, "The effect of patterns on thermal stress during rapid thermal processing of silicon wafers," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 1, pp. 99–107, Feb. 1998.

[19] R. Deaton and H. Massoud, "Manufacturability of rapid thermal oxidation of silicon: oxide thickness, oxide thickness variation, and system dependency," *IEEE Trans. Semicond. Manuf.*, vol. 5, no. 4, pp. 347–358, Nov. 1992.

[20] J. Sali, S. Patil, S. Jadkar, and M. Takwale, "Hot-wire CVD growth simulation for thickness uniformity," *Thin Solid Films*, vol. 395, nos. 1–2, pp. 66–70, Sep. 2001.

[21] P. Friedberg, "Spatial modeling of gate length variation for process-design co-optimization," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, 2007.

[22] W. Mann, F. Taber, P. Seitzer, and J. Broz, "The leading edge of production wafer probe test technology," in *Proc. Int. Test Conf.*, 2004, pp. 1168–1195.

[23] K. Balakrishnan and D. Boning, "Measurement and analysis of contact plug resistance variability," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2009, pp. 416–422.

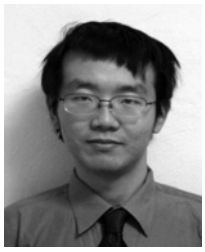
[24] S. Searle, G. Casella, and C. McCulloch, *Variance Components*. New York, USA: Wiley, 1992.

[25] C. Bishop, *Pattern Recognition and Machine Learning*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.

[26] R. Gonzalez and R. Woods, *Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.

[27] R. Maronna, R. Martin, and V. Yohai, *Robust Statistics: Theory and Methods*. New York, USA: Wiley, 2006.

[28] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*. Cambridge, MA, USA: Cambridge Univ. Press, 2007.



Wangyang Zhang (S'10–M'13) received the B.S. and M.S. degrees in computer science from Tsinghua University, Beijing, China, in 2008 and 2006, respectively, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2012.

He is currently with Cadence Design Systems, Inc., Pittsburgh.

Dr. Zhang was a recipient of the Best Paper Award from DAC in 2010.

Karthik Balakrishnan received the B.S. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2004, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA in 2006 and 2011, respectively.

He is currently a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

Xin Li (S'01–M'06–SM'10) received the Ph.D. degree in Electrical and Computer Engineering (ECE) from Carnegie Mellon University (CMU), Pittsburgh, PA, USA, in 2005.

He is currently an Assistant Professor with the ECE Department, CMU. His current research interests include computer-aided design and neural signal processing.

Dr. Li was a recipient of the NSF Faculty Early Career Development Award in 2012.

Duane S. Boning (S'90–M'91–SM'00–F'05) received the B.S., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.

He is currently a Professor of electrical engineering and computer science with MIT. His current research interests include variation modeling, control, and environmental issues in semiconductor and MEMS manufacturing.

Sharad Saxena (S'87–M'90–S'91–SM'97) received the Ph.D. degree in computer science from the University of Massachusetts, Amherst, MA, USA.

He is currently a Fellow with PDF Solutions, Inc., Richardson, TX, USA, where his responsibilities include developing methods and techniques for characterizing and modeling transistor performance and variation in advanced technologies.

Andrzej J. Strojwas (F'90) is currently a Joseph F. and Nancy Keithley Professor of Electrical and Computer Engineering with Carnegie Mellon University, Pittsburgh, PA, USA. Since 1997, he has served as the Chief Technologist with PDF Solutions, Inc., Pittsburgh.

Prof. Strojwas was a recipient of multiple awards for the best papers published in the IEEE TRANSACTIONS ON TRANSACTIONS ON COMPUTER-AIDED DESIGN, the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, and DAC.

Rob A. Rutenbar (S'77–M'84–SM'90–F'98) received the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1984.

He is currently the Abel Bliss Professor of engineering and the Head of the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA. His current research interests include today's commercial analog circuit and layout synthesis technology.

Dr. Rutenbar was a recipient of many awards, including several Best Paper Awards (e.g., DAC 1987, 2002, and 2010), the Semiconductor Research Corporation Aristotle Award for Excellence in Education in 2001, and the IEEE Circuits and Systems Industrial Pioneer Award in 2007. He is a fellow of the ACM.