Toward Efficient Spatial Variation Decomposition via Sparse Regression

Wangyang Zhang[1], Karthik Balakrishnan[2], Xin Li[1], Duane Boning[2] and Rob Rutenbar[3]
[1]Carnegie Mellon University, Pittsburgh, PA 15213, wangyan1@ece.cmu.edu, xinli@ece.cmu.edu
[2]Massachusetts Institute of Technology, Cambridge, MA 02139, karthikb@mit.edu, boning@mtl.mit.edu
[3]University of Illinois at Urbana-Champaign, Urbana, IL 61801, rutenbar@illinois.edu

ABSTRACT

In this paper, we propose a new technique to accurately decompose process variation into two different components: (1) spatially correlated variation, and (2) uncorrelated random variation. Such variation decomposition is important to identify systematic variation patterns at wafer and/or chip level for process modeling, control and diagnosis. We demonstrate that spatially correlated variation carries a unique sparse signature in frequency domain. Based upon this observation, an efficient sparse regression algorithm is applied to accurately separate spatially correlated variation from uncorrelated random variation. An important contribution of this paper is to develop a fast numerical algorithm that reduces the computational time of sparse regression by several orders of magnitude over the traditional implementation. Our experimental results based on silicon measurement data demonstrate that the proposed sparse regression technique can capture spatially correlated variation patterns with high accuracy. The estimation error is reduced by more than 3.5× compared to other traditional methods.

1. INTRODUCTION

With the continued scaling of CMOS technology, process variation has become a critical issue for design and manufacture of integrated circuits [1]-[4]. 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 in order to fully understand the variation sources and, hence, facilitate robust circuit design to achieve high parametric yield [15].

Towards this goal, identifying and modeling systematic variation patterns is of great importance. Once the systematic variation sources are found, it is possible to optimize the manufacturing process and/or modify the circuit design to improve yield. Traditionally, systematic variation patterns are often determined by calculating the averaged variation from a large number of wafers and/or chips [3]-[4]. As such, uncorrelated random variation can be statistically removed. These traditional approaches, however, suffer from several major limitations. First, it requires a large number of measured wafers/chips to accurately eliminate the impact of uncorrelated random variation. In practice, the number of available wafers/chips can be limited (e.g., in low-volume production). Second, the systematic variation patterns must be identical for all tested wafers/chips. If a number of wafers/chips carry a different systematic variation pattern (e.g., due to manufacturing equipment drift) or contain a lot of missing data (e.g., due to measurement error), they can substantially bias the estimation result.

It has been demonstrated in the literature that systematic variation often presents a unique spatial pattern [3]. Namely, systematic variation is spatially correlated. For example, it has been observed in [5] that the spatial correlation in gate length is partially caused by the systematic variation due to lithography. Motivated by these observations, we propose a new technique to uncover systematic variation patterns by decomposing process variation into two different components: (1) spatially correlated variation, and (2) uncorrelated random variation. In other words, by removing the uncorrelated random variation component, the remaining spatially correlated variation will accurately represent the systematic variation of interest.

Our proposed technique is based upon an important fact that spatially correlated variation and uncorrelated random variation present completely different signatures in frequency domain. Namely, spatially correlated variation typically carries a unique sparse structure in frequency domain [6]-[8], implying that it can be accurately represented by a small number of dominant DCT (i.e., discrete cosine transform) coefficients. On the other hand, uncorrelated random variation has a white frequency spectrum and the corresponding DCT coefficients are evenly distributed over all frequencies. By exploring the unique sparsity in frequency domain, we derive a sparse regression formulation to identify the dominant frequency-domain components and, hence, approximate the spatially correlated systematic variation.

Another important contribution of this paper is to borrow the Simultaneous Orthogonal Matching Pursuit (S-OMP) method from the statistics community [11] to solve the aforementioned sparse regression problem. A number of implementation details are carefully considered in order to further tune the S-OMP method to fit the need of our specific application. In particular, several new numerical algorithms are developed and integrated with S-OMP to substantially reduce the computational time for large-scale wafer/chip-level data analysis. Our key idea is to explore the special properties of DCT (e.g., orthogonality of DCT basis functions) [16] to simplify the numerical operations that are required by S-OMP.

The proposed variation decomposition technique has been validated by using the measurement data of contact plug resistance collected from 24 test chips in a 90 nm CMOS process. As will be demonstrated by the experimental results in Section 5, the proposed sparse regression approach reduces the estimation error by more than 3.5× compared to other traditional methods. In addition, our improved S-OMP algorithm achieves more than 600× speed-up over the traditional implementation.

The remainder of this paper is organized as follows. In Section 2, we first derive the mathematical formulation for the proposed variation decomposition problem and then describe the S-OMP algorithm in Section 3. Next, we develop several fast numerical algorithms to implement S-OMP in Section 4. The efficacy of the proposed method is demonstrated by several examples in Section 5. Finally, we conclude in Section 6.

2. VARIATION DECOMPOSITION

Let $g(x, y)$ be a two-dimensional function representing the spatial variation of interest, where x and y denote the coordinate of

978-1-4577-1400-9/11/$26.00 ©2011 IEEE

a spatial location within the two-dimensional plane. The spatial variation g can be the device-level threshold voltage variation within a chip, chip-level leakage current variation on a wafer, etc. In practice, the spatial variation g is measured at a finite number of spatial locations. Therefore, 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 [6]-[8]. If the spatial variation g is measured for multiple chips and/or wafers, it can be represented by a set of two-dimensional functions: $\{g_{(l)}(x, y);\ l = 1,2,\ldots L\}$, where L denotes the total number of wafers/chips. In this paper, we aim to decompose each spatial variation function $g_{(l)}(x, y)$ into two different components:

$$g_{(l)}(x,y) = s_{(l)}(x,y) + r_{(l)}(x,y) \quad (l = 1,2,\cdots,L) \tag{1}$$

where $\{s_{(l)}(x, y);\ x = 1,2,...,P,\ y = 1,2,...,Q\}$ and $\{r_{(l)}(x, y);\ x = 1,2,...,P,\ y = 1,2,...,Q\}$ stand for the spatially correlated variation and the uncorrelated random variation, respectively.

As demonstrated in [6]-[8], the spatial variation $\{g_{(l)}(x, y);\ x = 1,2,...,P,\ y = 1,2,...,Q\}$ can be mapped to frequency domain by a two-dimensional linear transform such as discrete cosine transform (DCT) [16]:

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

where

$$\alpha_u = \begin{cases} \sqrt{1/P} & (u=1) \\ \sqrt{2/P} & (2 \le u \le P) \end{cases} \tag{3}$$

$$\beta_v = \begin{cases} \sqrt{1/Q} & (v=1) \\ \sqrt{2/Q} & (2 \le v \le Q) \end{cases}. \tag{4}$$

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

$$g_{(l)}(x,y) = \sum_{u=1}^{P}\sum_{v=1}^{Q} \alpha_u \cdot \beta_v \cdot G_{(l)}(u,v) \cdot \cos\frac{\pi(2x-1)(u-1)}{2\cdot P} \cdot \cos\frac{\pi(2y-1)(v-1)}{2\cdot Q} \quad (l = 1,2,\cdots,L) \tag{5}$$

Due to the linearity of DCT [16], the DCT coefficients $\{G_{(l)}(u, v);\ u = 1,2,...,P,\ v = 1,2,...,Q\}$ can be further decomposed into two different components:

$$G_{(l)}(u,v) = S_{(l)}(u,v) + R_{(l)}(u,v) \quad (l = 1,2,\cdots,L) \tag{6}$$

where $\{S_{(l)}(u, v);\ u = 1,2,...,P,\ v = 1,2,...,Q\}$ and $\{R_{(l)}(u, v);\ u = 1,2,...,P,\ v = 1,2,...,Q\}$ denote the DCT coefficients of the spatially correlated variation $s_{(l)}(x, y)$ and the uncorrelated random variation $r_{(l)}(x, y)$ defined in (1). 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, similar to the case in (5).

To accurately solve the decomposition problem in (6), we first need to analyze the signatures of $S_{(l)}(u, v)$ and $R_{(l)}(u, v)$ in the DCT domain. As is demonstrated in [6]-[8], 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, $\lambda_{(l)}$ where $\lambda_{(l)} << PQ$) dominant DCT coefficients to satisfy:

$$\sum_{(u,v)\in D_{(l)}} S_{(l)}^2(u,v) \approx \sum_{u=1}^{P}\sum_{v=1}^{Q} S_{(l)}^2(u,v) \tag{7}$$

where $D_{(l)}$ denotes the set of the indices of the dominant DCT coefficients for $S_{(l)}(u, v)$. Eq. (7) 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 $\{S_{(l)}(u, v);\ (u, v) \in D_{(l)}\}$.

On the other hand, uncorrelated random variation can be characterized as white noise [17] and evenly distributed among all frequencies. Therefore, given the set of indices $D_{(l)}$, the following equation holds:

$$\sum_{(u,v)\in D_{(l)}} R_{(l)}^2(u,v) \approx \frac{\lambda_{(l)}}{PQ} \cdot \sum_{u=1}^{P}\sum_{v=1}^{Q} R_{(l)}^2(u,v). \tag{8}$$

Because of the inequality $\lambda_{(l)} << PQ$, we have $\lambda_{(l)}/PQ << 1$ in (8). If the value of $\lambda_{(l)}$ is sufficiently small (i.e., the DCT coefficients of spatially correlated variation are sufficiently sparse), the left-hand side of (8) is approximately zero and the following inequality holds:

$$\sum_{(u,v)\in D_{(l)}} R_{(l)}^2(u,v) << \sum_{(u,v)\notin D_{(l)}} S_{(l)}^2(u,v). \tag{9}$$

Based on these assumptions, an accurate approximation of the DCT coefficients $S_{(l)}(u, v)$ (corresponding to spatially correlated variation) can be expressed as:

$$\widetilde{S}_{(l)}(u,v) = \begin{cases} G_{(l)}(u,v) & ((u,v)\in D_{(l)}) \\ 0 & (otherwise) \end{cases}. \tag{10}$$

In other words, we simply approximate $S_{(l)}(u, v)$ by the dominant DCT coefficients $\{G_{(l)}(u, v);\ (u, v) \in D_{(l)}\}$.Comparing (6) and (10), it can be further proven that the approximation error of (10) is given by:

$$\sum_{u=1}^{P}\sum_{v=1}^{Q}\left[S_{(l)}(u,v) - \widetilde{S}_{(l)}(u,v)\right]^2 = \sum_{(u,v)\in D_{(l)}} R_{(l)}^2(u,v) + \sum_{(u,v)\notin D_{(l)}} S_{(l)}^2(u,v) \tag{11}$$

Given the assumptions in (7) and (9), the error terms in (11) are almost negligible.

In practice, however, Eq. (10) cannot be directly used for variation decomposition because of two reasons. First, the index set of dominant DCT coefficients $D_{(l)}$ is not known in advance and it must be estimated from the measurement data. Second, if the spatial variation $g_{(l)}(x, y)$ is not measured at all locations, it is not possible to directly calculate $G_{(l)}(u, v)$ from (2). In many practical applications, measurement error and manufacturing defect can result in missing data at a number of spatial locations, as is demonstrated in the literature [3], [8]. These observations, hence, motivate us to derive an efficient Simultaneous Orthogonal Matching Pursuit (S-OMP) algorithm [11] to determine $\tilde{S}_{(l)}(u, v)$ in (10) for accurate variation decomposition.

3. S-OMP ALGORITHM

In this section, we describe the S-OMP algorithm in detail. To this end, we first show a simplified version of S-OMP, referred to as Orthogonal Matching Pursuit (OMP) [12], where only the measurement data from a single wafer/chip are considered. Next, we derive the full S-OMP algorithm [11] that explores the correlation among multiple wafers/chips to further improve the accuracy for variation decomposition.

3.1 Orthogonal Matching Pursuit

The objective of OMP is to determine the index set $D_{(l)}$ so that a small number of dominant DCT coefficients can be identified to approximate the spatially correlated variation in (10). Mathematically, our variation decomposition problem can be formulated as the following optimization:

$$\begin{aligned} \underset{\eta_{(l)}}{\text{minimize}} \quad & \left\| A_{(l)} \cdot \eta_{(l)} - B_{(l)} \right\|_2^2 \\ \text{subject to} \quad & \left\| \eta_{(l)} \right\|_0 \le \lambda_{(l)} \end{aligned} \tag{12}$$

where $\|\bullet\|_2$ and $\|\bullet\|_0$ stand for the L_2-norm (i.e., the square root of the summation of the squares of all elements) and the L_0-norm (i.e., the number of non-zero elements) of a vector respectively, and:

$$A_{(l)} = \begin{bmatrix} A_{(l),1,1,1} & A_{(l),1,1,2} & \cdots & A_{(l),1,P,Q} \\ A_{(l),2,1,1} & A_{(l),2,1,2} & \cdots & A_{(l),2,P,Q} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(l),M_{(l)},1,1} & A_{(l),M_{(l)},1,2} & \cdots & A_{(l),M_{(l)},P,Q} \end{bmatrix} \tag{13}$$

$$\begin{aligned} A_{(l),m,u,v} = \alpha_u \cdot \beta_v \cdot \cos \frac{\pi\left(2x_{(l),m} - 1\right) \cdot (u-1)}{2 \cdot P} \\ \cdot \cos \frac{\pi\left(2y_{(l),m} - 1\right) \cdot (v-1)}{2 \cdot Q} \end{aligned} \tag{14}$$

$$\eta_{(l)} = \left[\widetilde{S}_{(l)}(1,1) \quad \widetilde{S}_{(l)}(1,2) \quad \cdots \quad \widetilde{S}_{(l)}(P,Q) \right]^T \tag{15}$$

$$B_{(l)} = \left[g_{(l)}\left(x_{(l),1}, y_{(l),1}\right) \quad \cdots \quad g_{(l)}\left(x_{(l),M_{(l)}}, y_{(l),M_{(l)}}\right) \right]^T . \tag{16}$$

In (12)-(16), the vector $B_{(l)}$ represents the measurement data collected from $M_{(l)}$ different spatial locations $\{(x_{(l),m}, y_{(l),m})$; $m = 1,2,\ldots,M_{(l)}\}$ of the lth wafer/chip, the vector $\eta_{(l)}$ contains the unknown DCT coefficients for the spatially correlated variation that we want to extract, and the matrix $A_{(l)}$ defines the linear transform to map the DCT coefficients from the frequency domain to the spatial domain. The optimization in (12) attempts to use a small number of (i.e., $\lambda_{(l)}$) dominant DCT coefficients to approximate the measurement data $B_{(l)}$ with least-squares error.

Studying (12), we would have two important observations. First, if the number of measured samples (i.e., $M_{(l)}$) is equal to the total number of DCT coefficients (i.e., PQ), the matrix $A_{(l)}$ represents the IDCT matrix and it is a full-rank square matrix. On the other hand, if $M_{(l)}$ is less than PQ (e.g., due to missing data), the matrix $A_{(l)}$ contains $M_{(l)}$ rows taken from the IDCT matrix and it is not simply a square matrix.

In general, solving the optimization in (12) is not trivial, since the problem is NP-hard. OMP [12] is an efficient greedy algorithm to approximate the solution of (12). It was recently adopted by the CAD community for large-scale performance modeling [9]. In this paper, we further extend the OMP algorithm to our application of variation decomposition. In what follows, we briefly review the major steps of the OMP algorithm. More details on OMP can be found in the literature [9], [12].

The key idea of OMP is to iteratively use the inner product to identify a small number of important DCT coefficients. Towards this goal, we re-write the matrix $A_{(l)}$ by its column vectors:

$$A_{(l)} = \left[A_{(l),1} \quad A_{(l),2} \quad \cdots \quad A_{(l),PQ} \right] \tag{17}$$

where each column vector $A_{(l),i}$ can be conceptually viewed as a basis vector associated with the DCT coefficient $\eta_{(l),i}$. 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 DCT coefficient $\eta_{(l),i}$) is an important component to approximate $B_{(l)}$.

Based on this idea, OMP applies an iterative process to find a set of important basis vectors, as summarized in Algorithm 1. At each iteration, OMP performs two major operations. First, it selects the basis vector $A_{(l),s}$ that is most "correlated" to the residual $Res_{(l)}$. Second, the DCT coefficients associated with all selected basis vectors are solved by least-squares fitting.

It should be noted that Algorithm 1 relies on a given input parameter $\lambda_{(l)}$. In practice, the value of $\lambda_{(l)}$ is not known in advance. However, it can be accurately estimated by cross-validation, as will be discussed in detail in Section 3.3.

Algorithm 1: Orthogonal Matching Pursuit (OMP)

1. Start from the optimization problem in (12) with a given integer $\lambda_{(l)}$ specifying the total number of basis vectors.
2. Initialize the residual $Res_{(l)} = B_{(l)}$, the set $\Omega_{(l)} = \{\}$, and the iteration index $p = 1$.
3. Select the new basis vector $A_{(l),s}$ according to the following criterion:
$$\underset{s}{\text{maximize}} \quad \left| \left\langle Res_{(l)}, A_{(l),s} \right\rangle \right| . \tag{18}$$
4. Update $\Omega_{(l)}$ by $\Omega_{(l)} = \Omega_{(l)} \cup \{s\}$.
5. Solve the least-squares fitting:
$$\underset{\eta_{(l),i}, i \in \Omega_{(l)}}{\text{minimize}} \quad \left\| \sum_{i \in \Omega_{(l)}} A_{(l),i} \cdot \eta_{(l),i} - B_{(l)} \right\|_2^2 . \tag{19}$$
6. Calculate the residual:
$$Res_{(l)} = B_{(l)} - \sum_{i \in \Omega_{(l)}} A_{(l),i} \cdot \eta_{(l),i} . \tag{20}$$
7. If $p < \lambda_{(l)}$, $p = p + 1$ and go to Step 3.
8. For any $i \notin \Omega_{(l)}$, set $\eta_{(l),i} = 0$.

3.2 Simultaneous OMP

While the spatially correlated variation for multiple wafers/chips can be extracted by independently performing OMP, this method is clearly not optimal since it ignores the strong correlation among different wafers/chips. Such strong correlation exists, if these wafers/chips are produced by the same manufacturing line and, hence, a significant portion of systematic variation can be shared [8]. In this sub-section, we further extend the OMP algorithm and derive an efficient Simultaneous Orthogonal Matching Pursuit (S-OMP) algorithm [11] so that the aforementioned correlation information can be used to improve the accuracy of variation decomposition.

As demonstrated in [8], if multiple wafers share similar spatial variation patterns, the corresponding DCT coefficients are strongly correlated. In this case, dominant DCT coefficients can be found at a number of common frequencies shared by all wafers. A similar observation can be made at chip level, where the systematic variation of a chip is often characterized by layout-dependent patterns. Since multiple chips share the same layout design, their systematic variation is expected to share similar spatial patterns. Hence, the dominant DCT coefficients associated with chip-level systematic variation should be distributed over a set of common frequencies shared by multiple chips.

Based upon these observations, we propose to model the spatial variation of multiple wafers/chips by a shared index set D for dominant DCT coefficients. Namely, we assume:

$$D_{(1)} = D_{(2)} = \cdots = D_{(L)} = D . \tag{21}$$

Consequently, the sizes of the sets $\{D_{(l)}; l = 1,2,\ldots,L\}$, i.e., $\{\lambda_{(l)}; l = 1,2,\ldots,L\}$, are identical and can be modeled by a single parameter λ:

$$\lambda_{(1)} = \lambda_{(2)} = \cdots = \lambda_{(L)} = \lambda \ . \tag{22}$$

With (21)-(22) in mind, we re-visit the OMP algorithm (i.e., Algorithm 1) where a set of dominant DCT coefficients are selected to approximate the spatially correlated systematic variation. At each iteration of Algorithm 1, a single DCT basis vector is chosen according to the inner product in (18). For S-OMP, since the index set of dominant DCT coefficients is shared for L different wafers/chips as shown in (21), we use the linear combination of multiple inner products as a quantitative criterion for basis vector selection:

$$\underset{s}{\text{maximize}} \quad \sum_{l=1}^{L} \left| \left\langle Res_{(l)}, A_{(l),s} \right\rangle \right| . \tag{23}$$

Eq. (23) is expected to be more accurate than (18), 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/chips, the impact of random noise is reduced and the spatial pattern associated with systematic variation can be accurately detected. This is the fundamental reason why S-OMP is preferred over OMP, if the spatially correlated systematic variation shares similar patterns across multiple wafers/chips. Algorithm 2 summarizes the major steps of the aforementioned S-OMP algorithm. Note that S-OMP is an extended version of OMP (i.e. Algorithm 1). If there is only one wafer/chip (i.e., $L = 1$), S-OMP is exactly equivalent to OMP.

Algorithm 2: Simultaneous OMP (S-OMP)

1. Start from the optimization problem in (12) for L wafers/chips $l \in \{1,2,\ldots,L\}$ with a given integer λ specifying the total number of basis vectors.
2. Initialize the set $\Omega = \{\}$, and the iteration index $p = 1$.
3. For each $l \in \{1,2,\ldots,L\}$, set the residual $Res_{(l)} = B_{(l)}$.
4. Select the new basis vector s according to (23).
5. Update Ω by $\Omega = \Omega \cup \{s\}$.
6. For each $l \in \{1,2,\ldots,L\}$, solve the least-squares fitting in (19).
7. Calculate the residual for $l \in \{1,2,\ldots,L\}$ by using (20).
8. If $p < \lambda$, $p = p + 1$ and go to Step 4.
9. For any $i \notin \Omega$, set $\eta_{(l),i} = 0$.

3.3 Cross-Validation

The S-OMP algorithm (i.e., Algorithm 2) relies on a user defined parameter λ to control the number of dominant DCT coefficients that should be selected. In practice, λ is not known in advance. The appropriate value of λ must be determined by considering the following two important issues. First, if λ is too small, S-OMP cannot select a sufficient number of basis vectors to represent the spatially correlated variation, thereby leading to large modeling error. On the other hand, if λ is too large, S-OMP can incorrectly select too many DCT coefficients and some of these coefficients are associated with uncorrelated random variation, instead of spatially correlated systematic variation. It, again, results in large modeling error due to over-fitting. In order to achieve the best accuracy, we must accurately estimate the modeling error for different λ values and then find the optimal λ with minimum error.

In this paper, we adopt the cross-validation method [18] to estimate the modeling error for our variation decomposition application. An F-fold cross-validation partitions the entire data set into F groups. Modeling error is estimated according to the cost function in (12) from F independent runs. In each run, one of the F groups is used to estimate the modeling error and all other groups are used to calculate the DCT coefficients. Note that the training data for coefficient estimation and testing data for error estimation are not overlapped. Hence, over-fitting can be easily detected. In addition, different groups should be selected for error estimation in different runs. As such, each run results in an error value ε_f ($f = 1,2,...,F$) that is measured from a unique group of data points. The final modeling error is computed as the average of $\{\varepsilon_f; f = 1, 2,...,F\}$, i.e., $\varepsilon = (\varepsilon_1 + \varepsilon_2 + ... + \varepsilon_F)/F$.

4. IMPLEMENTATION DETAILS

While Algorithm 2 summarizes the major steps of S-OMP for variation decomposition, a number of implementation details must be carefully considered in order to make the S-OMP algorithm computationally efficient for large-scale problems. In this section, we derive several efficient numerical algorithms to address the aforementioned issue related to computational cost.

4.1 Inner Product Computation

It can be easily observed from Algorithm 2 that the computational cost is dominated by two steps: the inner product computation in Step 4 and the least-squares fitting in Step 6. In this sub-section, we first derive an efficient numerical algorithm to calculate the inner product values in (23). We will discuss the numerical algorithm for least-squares fitting in the next sub-section.

In order to appropriately select the basis vectors by (23), the inner product $<Res_{(l)}, A_{(l),i}>$ must be calculated for all basis vectors $i \in \{1,2,\ldots,PQ\}$ and all wafers/chips $l \in \{1,2,\ldots,L\}$. If the inner product values are simply calculated by vector-vector multiplications, the computational cost is in the order of $O(LP^2Q^2)$. Note that the computational cost quadratically increases with the problem size PQ. Hence, the aforementioned implementation can quickly become computationally intractable, as the problem size increases.

For this reason, an efficient numerical algorithm for inner product computation is needed in order to reduce the computational cost. Towards this goal, we first re-write the inner product $<Res_{(l)}, A_{(l),i}>$ as:

$$\left\langle Res_{(l)}, A_{(l),i} \right\rangle = A_{(l),i}^{T} \cdot Res_{(l)} \ . \tag{24}$$

For each $l \in \{1,2,\ldots,L\}$, we need to calculate (24) for each basis vector, i.e., $i \in \{1,2,\ldots,PQ\}$. The results can be expressed by the following matrix-vector multiplication:

$$\begin{bmatrix} \left\langle Res_{(l)}, A_{(l),1} \right\rangle \\ \left\langle Res_{(l)}, A_{(l),2} \right\rangle \\ \vdots \\ \left\langle Res_{(l)}, A_{(l),PQ} \right\rangle \end{bmatrix} = A_{(l)}^{T} \cdot Res_{(l)} . \tag{25}$$

In other words, by calculating the matrix-vector multiplication in (25), we are able to obtain the inner product values for all (i.e., PQ) basis vectors.

If the measurement of the lth wafer/chip does not contain any missing data, the matrix $A_{(l)}$ in (25) represents the IDCT matrix and it is a full-rank square matrix, as defined in (13). In this case, since DCT/IDCT is an orthogonal transform [16], $A_{(l)}^{T} = A_{(l)}^{-1}$ is exactly the DCT matrix. Namely, calculating the inner product values in (25) is equivalent to performing DCT on the residual $Res_{(l)}$. Similar to fast Fourier transform (FFT), there exists a number of fast algorithms for DCT/IDCT. The computational cost

of these fast algorithms is in the order of $O(PQ \cdot \log(PQ))$ [16]. Therefore, by using a fast DCT algorithm, the computational cost for Step 4 of Algorithm 2 is reduced from $O(LP^2Q^2)$ to $O(LPQ \cdot \log(PQ))$.

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

$$Res^*_{(l)} = W_{(l)} \cdot \begin{bmatrix} Res_{(l)} \\ 0 \end{bmatrix} \tag{26}$$

where $W_{(l)}$ is a permutation matrix to map the residual $Res_{(l)}$ and the zero vector to the appropriate elements in $Res^*_{(l)}$.

Applying DCT to the augmented vector $Res^*_{(l)}$ yields:

$$A^{*T} \cdot Res^*_{(l)} = A^{*T} \cdot W_{(l)} \cdot \begin{bmatrix} Res_{(l)} \\ 0 \end{bmatrix} \tag{27}$$

where A^* represents the IDCT matrix and, hence, A^{*T} is the DCT matrix. Remember that the matrix $A_{(l)}$ in (13) contains $M_{(l)}$ rows taken from the IDCT matrix A^*. Hence, the matrix $A^{*T} \cdot W_{(l)}$ in (27) can be re-written as:

$$A^{*T} \cdot W_{(l)} = \begin{bmatrix} A^T_{(l)} & A^T_{(\tilde{l})} \end{bmatrix} \tag{28}$$

where the matrix $A_{(\tilde{l})}$ contains the $PQ - M_{(l)}$ rows of A^* that are not included in $A_{(l)}$ due to missing data. Substituting (28) into (27), we have:

$$A^{*T} \cdot Res^*_{(l)} = \begin{bmatrix} A^T_{(l)} & A^T_{(\tilde{l})} \end{bmatrix} \cdot \begin{bmatrix} Res_{(l)} \\ 0 \end{bmatrix} = A^T_{(l)} \cdot Res_{(l)} . \tag{29}$$

Note that the DCT results in (29) are exactly equal to the inner product values in (25). 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 2 is again reduced from $O(LP^2Q^2)$ to $O(LPQ \cdot \log(PQ))$.

In addition to the reduction in computational cost, the aforementioned fast algorithm based on DCT can also efficiently reduce the memory consumption. Note that the direct matrix-vector multiplication in (25) requires to explicitly form a dense matrix $A_{(l)}$ with about P^2Q^2 entries. While it is possible to calculate each inner product in (24) 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 2. 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 augmented vector $Res^*_{(l)}$ in (26) with PQ entries. A fast DCT algorithm can be applied to $Res^*_{(l)}$ without explicitly building the DCT matrix in memory, thereby significantly reducing the memory consumption for large-scale problems.

4.2 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 2. The goal is to solve the optimization problem in (19). In this sub-section, we will develop an efficient numerical algorithm to reduce the computational cost of (19).

We first re-write (19) for the lth wafer/chip at the pth iteration step:

$$\underset{\eta_{(l),(p)}}{\text{minimize}} \quad \left\| A_{(l),(p)} \cdot \eta_{(l),(p)} - B_{(l)} \right\|_2^2 \tag{30}$$

where the matrix $A_{(l),(p)}$ contains p column vectors selected from $A_{(l)}$ and the vector $\eta_{(l),(p)}$ contains the DCT coefficients corresponding to these selected basis vectors. The relation between $A_{(l),(p)}$ and $A_{(l)}$ can be further expressed as:

$$A_{(l)} \cdot W_{(p)} = \begin{bmatrix} A_{(l),(p)} & A_{(l),(\tilde{p})} \end{bmatrix} \tag{31}$$

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

The least-squares solution $\eta_{(l),(p)}$ of (30) satisfies the following normal equation [19]:

$$A^T_{(l),(p)} \cdot A_{(l),(p)} \cdot \eta_{(l),(p)} = A^T_{(l),(p)} \cdot B_{(l)} . \tag{32}$$

Traditionally, the solution $\eta_{(l),(p)}$ of (32) is solved by QR decomposition [19]:

$$A_{(l),(p)} = Q_{(l),(p)} \cdot R_{(l),(p)} \tag{33}$$

where $Q_{(l),(p)}$ is an $M_{(l)}$-by-p matrix with orthonormal columns and $R_{(l),(p)}$ is a p-by-p upper triangular matrix. Substituting (33) into (32) yields:

$$R_{(l),(p)} \cdot \eta_{(l),(p)} = Q^T_{(l),(p)} \cdot B_{(l)} . \tag{34}$$

In (34), since $R_{(l),(p)}$ is upper triangular, $\eta_{(l),(p)}$ can be solved by back substitution. The computational cost of the aforementioned least-squares fitting is dominated by the QR decomposition step and it is in the order of $O(M_{(l)} \cdot p^2)$.

The traditional least-squares solver based on QR decomposition is not computationally efficient for large-scale problems. An alternative way to solve (30) is based on an iterative algorithm that is referred to as the LSQR method [13]. LSQR relies on the bi-diagonalization process of the matrix $A_{(l),(p)}$. During its iterations, LSQR generates a sequence of solutions to approximate $\eta_{(l),(p)}$. These solutions are exactly identical to the results calculated by the conjugate gradient method [19] for the normal equation in (32). However, unlike the conjugate gradient method that suffers from numerical issues when solving (32), LSQR aims to directly solve (30) in order to improve numerical stability. The details of LSQR can be found in [13].

When applying LSQR, it is not necessary to explicitly form the matrix $A_{(l),(p)}$. Instead, only the matrix-vector multiplications $A_{(l),(p)} \cdot \alpha$ and $A_{(l),(p)}{}^T \cdot \beta$, where α is a p-by-1 vector and β is an $M_{(l)}$-by-1 vector, are required. These matrix-vector multiplications can be efficiently calculated by applying a fast numerical algorithm. In what follows, we will show the mathematical formulation of our proposed fast algorithm.

First, to efficiently compute $A_{(l),(p)} \cdot \alpha$, we construct an augmented vector $\alpha^* \in R^{PQ}$:

$$\alpha^* = W_{(p)} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \tag{35}$$

where $W_{(p)}$ is the permutation matrix defined in (31). We conceptually consider the augmented vector α^* as a set of DCT coefficients and apply IDCT to it:

$$A^* \cdot \alpha^* = A^* \cdot W_{(p)} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \tag{36}$$

where A^* denotes the IDCT matrix as defined in (27). On the other hand, we can derive the following equation from (28):

$$A^* = W_{(l)} \cdot \begin{bmatrix} A_{(l)} \\ A_{(\tilde{l})} \end{bmatrix} . \tag{37}$$

Substituting (37) into (36) yields:

$$A^* \cdot \alpha^* = W_{(l)} \cdot \begin{bmatrix} A_{(l)} \cdot W_{(p)} \\ A_{(\tilde{l})} \cdot W_{(p)} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix}. \quad (38)$$

In (38), $A_{(l)} \cdot W_{(p)}$ can be represented as two sub-matrices as shown in (31). If we similarly re-write $A_{(\tilde{l})} \cdot W_{(p)}$ as two sub-matrices:

$$A_{(\tilde{l})} \cdot W_{(p)} = \begin{bmatrix} A_{(\tilde{l}),(p)} & A_{(\tilde{l}),(\tilde{p})} \end{bmatrix} \quad (39)$$

Eq. (38) becomes:

$$A^* \cdot \alpha^* = W_{(l)} \cdot \begin{bmatrix} A_{(l),(p)} & A_{(l),(\tilde{p})} \\ A_{(\tilde{l}),(p)} & A_{(\tilde{l}),(\tilde{p})} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = W_{(l)} \cdot \begin{bmatrix} A_{(l),(p)} \cdot \alpha \\ A_{(\tilde{l}),(p)} \cdot \alpha \end{bmatrix}. \quad (40)$$

Since $W_{(l)}$ is a permutation matrix, Eq. (40) is equivalent to:

$$\begin{bmatrix} A_{(l),(p)} \cdot \alpha \\ A_{(\tilde{l}),(p)} \cdot \alpha \end{bmatrix} = W_{(l)}^T \cdot A^* \cdot \alpha^*. \quad (41)$$

Eq. (41) reveals an important fact that the matrix-vector multiplication $A_{(l),(p)} \cdot \alpha$ can be efficiently computed by applying IDCT to the augmented vector α^*. The value of $A_{(l),(p)} \cdot \alpha$ is determined by selecting the appropriate elements from the IDCT result $A^* \cdot \alpha^*$. If a fast IDCT algorithm is applied [16], the computational cost of the aforementioned matrix-vector calculation is in the order of $O(PQ \cdot \log(PQ))$.

Next, we consider the other matrix-vector multiplication $A_{(l),(p)}^T \cdot \beta$ that is required by the LSQR algorithm. Similarly, we first construct an augmented vector $\beta^* \in R^{PQ}$:

$$\beta^* = W_{(l)} \cdot \begin{bmatrix} \beta \\ 0 \end{bmatrix} \quad (42)$$

where $W_{(l)}$ is the permutation matrix defined in (26). We apply DCT to the augmented vector β^*:

$$A^{*T} \cdot \beta^* = A^{*T} \cdot W_{(l)} \cdot \begin{bmatrix} \beta \\ 0 \end{bmatrix} \quad (43)$$

where A^{*T} is the DCT matrix as defined in (27). Substituting (37) into (43) yields:

$$A^{*T} \cdot \beta^* = \begin{bmatrix} A_{(l)}^T & A_{(\tilde{l})}^T \end{bmatrix} \cdot W_{(l)}^T \cdot W_{(l)} \cdot \begin{bmatrix} \beta \\ 0 \end{bmatrix} = A_{(l)}^T \cdot \beta. \quad (44)$$

Based on (31), Eq. (44) can be further re-written as:

$$\begin{bmatrix} A_{(l),(p)}^T \cdot \beta \\ A_{(l),(\tilde{p})}^T \cdot \beta \end{bmatrix} = W_{(p)}^T \cdot A^{*T} \cdot \beta^*. \quad (45)$$

Hence, the matrix-vector multiplication $A_{(l),(p)}^T \cdot \beta$ can be calculated by applying DCT to the augmented vector β^*. The value of $A_{(l),(p)}^T \cdot \beta$ is determined by selecting the appropriate elements from the DCT result $A^{*T} \cdot \beta^*$. The computational cost is in the order of $O(PQ \cdot \log(PQ))$.

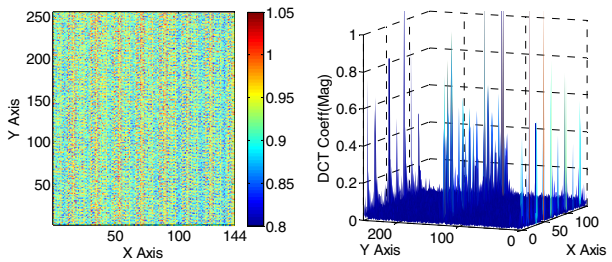
Finally, it is worth mentioning that similar to other iterative solvers, a good initial guess should be provided to LSQR to achieve fast convergence. If the initial guess is close to the actual solution, LSQR can reach convergence in a few iterations [13]. In this paper, LSQR is required at each iteration step of the S-OMP algorithm (i.e., Algorithm 2). When Algorithm 2 is applied, 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 2~3 iterations in our tested examples.

5. NUMERICAL EXAMPLES

In this section, we demonstrate the efficacy of our proposed variation decomposition algorithm using several examples. All numerical experiments are performed on a 2.8GHz Linux server.

5.1 Measurement Data for Contact Plug Resistance

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, as described in [10]. Among these 24 test chips, three of them contain missing data due to external measurement error. The number of failed measurements are 2936, 864 and 8 for these three chips, respectively.



(a) (b)

Figure 1. (a) Measured contact plug resistance (normalized) of a 144×256 array for one of the 24 test chips. (b) Discrete cosine transform (DCT) coefficients (magnitude) of the measured contact plug resistance for the same test chip.

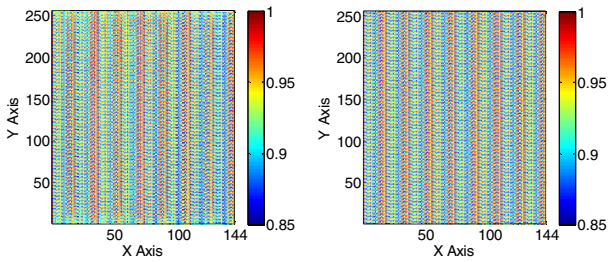
Figure 1(a) shows the measured contact plug resistance (normalized) from one of the 24 test chips. Studying Figure 1(a), we would notice that there is a unique spatial pattern due to layout dependency. However, the spatial pattern is not clearly visible because of the large-scale uncorrelated random variation found in this example.

Figure 1(b) further shows the DCT coefficients (magnitude) of the measured contact plug resistance for the same test chip. Note that there only exist a small number of dominant DCT coefficients with large magnitude. These DCT coefficients are distributed over a small number of frequencies, representing a unique signature of the layout-dependent systematic variation in frequency domain. All other DCT coefficients are small in magnitude and have a white frequency spectrum (i.e., evenly distributed over all frequencies). They correspond to the uncorrelated random variation that we observe from Figure 1(a). These observations demonstrate the important fact that the spatially correlated systematic variation can be extracted by identifying the dominant DCT coefficients in frequency domain.

A. Variation Decomposition

We apply the proposed S-OMP algorithm (i.e. Algorithm 2) to extract the layout-dependent systematic variation of all test chips. The extracted systematic variation of the chip in Figure 1(a) is shown in Figure 2(a). Comparing Figure 2(a) with Figure 1(a), we would notice that the spatial pattern of systematic variation becomes clear, after S-OMP is applied. Such a spatial variation pattern can serve as an important basis for diagnosing the sources of systematic variation.

In this example, the systematic variation is mainly caused by different layout patterns regularly distributed over the entire chip. To verify the layout dependency, we plot the spatial distribution of different layout patterns in Figure 2(b) where there exist 55 layout patterns in total and different layout patterns are shown in different colors. Note that Figure 2(b) perfectly matches Figure 2(a). It, in turn, demonstrates that the aforementioned layout dependency is the dominant source for the extracted systematic variation in Figure 2(a).



(a) (b)

Figure 2. (a) Extracted layout-dependent systematic variation (normalized) of contact plug resistance. (b) Spatial distribution of different contact layout patterns in the test chip.

B. *Runtime Comparison*

To demonstrate the efficiency of the fast numerical algorithms proposed in Section 4, we implement three different versions of OMP/S-OMP 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 (24) and the least-squares fitting is directly computed by the QR decomposition in (33)-(34). In the second implementation, the traditional inner product calculation is replaced by the fast algorithm proposed in Section 4.1. Finally, in the third implementation, both the inner product and the least-squares fitting are calculated by the fast algorithms proposed in Section 4.

For testing and comparison purposes, we first run the OMP algorithm (i.e., Algorithm 1) with the aforementioned three implementations. Table 1 shows the computational time for the proposed variation decomposition of a single test chip. Note that the fast algorithm for inner product computation achieves 73× speed-up and the fast least-squares fitting further brings 8.8× speed-up. The overall speed-up achieved by our proposed fast algorithms is 647×, compared to the traditional direct implementation.

Table 1. Computational time of variation decomposition for a single chip by OMP

Inner product	Least-squares fitting	CPU time (Sec.)
Direct	Direct	2.88×10^{6}
Fast	Direct	3.93×10^{4}
Fast	Fast	4.45×10^{3}

Table 2. Computational time of variation decomposition for 24 chips by S-OMP

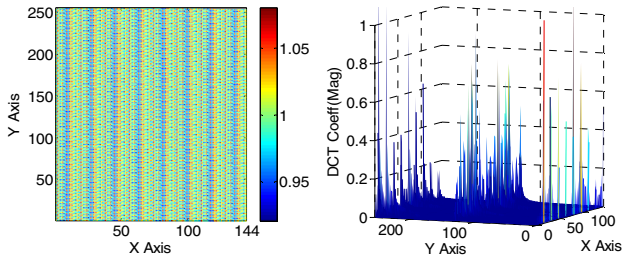
Inner Product	Least-squares fitting	CPU time (Sec.)
Fast	Direct	5.20×10^{6}
Fast	Fast	1.97×10^{5}

Next, we run S-OMP for all 24 test chips and Table 2 compares the computational time for two different implementations. Once S-OMP 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 2. In this example, the proposed fast algorithm for least-squares fitting achieves 26.3× speed-up over the direct implementation.

5.2 Synthetic Data for Contact Plug Resistance

To further validate the accuracy of the proposed S-OMP algorithm, we create a set of synthetic data for contact plug resistance. Similar to the silicon measurement data shown in Section 5.1, the synthetic data set also contains 24 test chips, with a 144×256 array of test structures in each chip.

Three different variation sources are modeled for the synthetic data set: (1) die-to-die variation (σ = 5%), (2) layout-dependent systematic variation (σ = 3.5%), (3) uncorrelated random variation (σ = 3.5%). The standard deviation of these variation sources is approximately equal to what is observed from the silicon measurement in Section 5.1. Since we exactly know the systematic variation for the synthetic data set, it enables us to quantitatively compare the accuracy of the proposed variation decomposition algorithm with several traditional techniques.



(a) (b)

Figure 3. (a) Layout-dependent systematic variation (normalized) of contact plug resistance for one of the 24 test chips in the synthetic data set. (b) Discrete cosine transform (DCT) coefficients (magnitude) of the systematic variation for the same synthetic test chip.

Figure 3(a) shows the layout-dependent systematic variation (normalized) of contact plug resistance for one of the 24 test chips in the synthetic data set. Figure 3(b) further shows the DCT coefficients (magnitude) of the systematic variation for the same synthetic test chip. Note that the DCT coefficients are sparse in frequency domain where most DCT coefficients are close to 0.

A. *Accuracy Comparison*

For testing and comparison purposes, we apply several different algorithms to extract the layout-dependent systematic variation in this example: (1) the proposed S-OMP algorithm (i.e., Algorithm 2), (2) the OMP algorithm (i.e., Algorithm 1), (3) the non-local means method [14], (4) the moving average method [16], (5) the Wiener filter method [16], (6) the Gaussian filter method [16], and (7) the wavelet thresholding method [16]. Except S-OMP and OMP, other methods are borrowed from the image processing community. To compare the accuracy of these different techniques, we define the average estimation error of systematic variation as:

$$Error = \frac{1}{24}\sum_{l=1}^{24}\sqrt{\frac{\sum_{x=1}^{144}\sum_{y=1}^{256}\left[s_{(l)}(x,y)-\tilde{s}_{(l)}(x,y)\right]^2}{\sum_{x=1}^{144}\sum_{y=1}^{256}\left[s_{(l)}(x,y)\right]^2}}. \quad (46)$$

where $s_{(l)}(x, y)$ and $\tilde{s}_{(l)}(x, y)$ denote the exact systematic variation and the estimated systematic variation for the lth chip, respectively.

Table 3 shows the average estimation error for seven different algorithms. Studying the results in Table 3, we would have two important observations. First, our proposed S-OMP algorithm is more accurate than the simple OMP algorithm. Compared to OMP, S-OMP improves the accuracy by exploring the correlation information among different chips, as discussed in Section 3.2.

Second, the proposed S-OMP algorithm achieves more than 3.5× error reduction over the traditional image processing techniques that have been widely applied for noise removal. Most image processing methods are particularly developed to capture the low-frequency components of a 2-D image (e.g., by local smoothing). They cannot accurately capture the high-frequency DCT coefficients shown in Figure 3(b), thereby resulting in large error.

Table 3. Average estimation error of layout-dependent systematic variation for seven different algorithms

Algorithm	Error
S-OMP (Algorithm 2)	0.67%
OMP (Algorithm 1)	1.00%
Non-local means [14]	2.44%
Moving average [16]	3.00%
Wiener filter [16]	3.11%
Gaussian filter [16]	2.55%
Wavelet thresholding [16]	3.06%

B. Missing Data

Finally, to further study the impact of missing data on our proposed S-OMP algorithm, we purposely introduce a number of missing data into the synthetic data set. The locations of these missing data are made identical to the silicon measurement data in Section 5.1. Namely, three chips contain missing data where the number of failed measurements are 2936, 864 and 8 respectively. With these missing data, the estimation error of S-OMP only increases by 0.01% for the chip with 2936 missing samples, and no notable change in estimation error is observed for other chips. Consequently, the change of average estimation error is almost negligible. In addition, we further randomly inject 10% and 20% missing samples to each synthetic chip. The average error of S-OMP is 0.68% and 0.72% in these two cases, respectively. These observations, in turn, demonstrate an important fact that the proposed S-OMP algorithm is extremely robust to the missing data caused by measurement error.

6. CONCLUSIONS

In this paper, we propose a new technique to efficiently separate spatially correlated systematic variation from uncorrelated random variation. The proposed method is based upon the fact that spatially correlated variation typically carries a unique sparse signature in frequency domain and it can be accurately represented by a small number of dominant DCT coefficients. An efficient S-OMP algorithm is borrowed from the statistics community to accurately find these dominant DCT coefficients corresponding to systematic variation. In addition, a number of fast numerical algorithms are developed to make the computational cost tractable for large-scale data analysis problems. Our experimental results for contact plug resistance demonstrate that the proposed S-OMP algorithm achieves more than 3.5× error reduction compared to other traditional methods. The variation decomposition technique developed by this paper can be applied to a number of practical applications, including manufacturing process modeling, control and diagnosis.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support of the C2S2 Focus Center and the Interconnect Focus Center, two of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity. This work is also supported in part by the National Science Foundation under contract CCF–0915912.

8. REFERENCES

[1] S. Nassif, "Delay variability: sources, impacts and trends," *IEEE ISSCC*, pp. 368-369, 2000.

[2] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2009.

[3] A. Gattiker, "Unraveling variability for process/product improvement," *IEEE ITC*, pp. 1-9, 2008.

[4] S. Reda and S. Nassif, "Accurate spatial estimation and decomposition techniques for variability characterization," *IEEE Trans. on Semiconductor Manufacturing*, vol. 23, no. 3, pp. 345-357, Aug. 2010.

[5] 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," *Proceedings of SPIE*, vol. 5756, pp. 178-188, May. 2005.

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

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

[8] 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," *IEEE ICCAD*, pp. 47-54, 2010.

[9] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits," *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1661-1668, Nov. 2010.

[10] K. Balakrishnan and D. Boning, "Measurement and analysis of contact plug resistance variability," *IEEE CICC*, pp. 416-422, 2009.

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

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

[13] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. on Mathematical Software*, vol. 8, no. 1, pp. 43-71, Mar. 1982.

[14] A. Buades, B. Coll, and J. Morel, "A nonlocal algorithm for image denoising," *IEEE CVPR*, vol. 2, 2005, pp. 60-65.

[15] M. Orshansky, S. Nassif, and D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach*, Springer, 2007.

[16] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2007.

[17] A. Oppenheim, *Signals and Systems*, Prentice Hall, 1996.

[18] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.

[19] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007.